

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4905637号
(P4905637)

(45) 発行日 平成24年3月28日(2012.3.28)

(24) 登録日 平成24年1月20日(2012.1.20)

(51) Int.Cl. F I
G O 6 N 3/00 (2006.01) G O 6 N 3/00 5 5 O E

請求項の数 4 (全 19 頁)

<p>(21) 出願番号 特願2005-292905 (P2005-292905) (22) 出願日 平成17年10月5日 (2005.10.5) (65) 公開番号 特開2007-102581 (P2007-102581A) (43) 公開日 平成19年4月19日 (2007.4.19) 審査請求日 平成20年3月27日 (2008.3.27)</p> <p>(出願人による申告)平成17年度独立行政法人情報通信研究機構、研究テーマ「人間情報コミュニケーションの研究開発」に関する委託研究、産業活力再生特別措置法第30条の適用を受ける特許出願</p>	<p>(73) 特許権者 393031586 株式会社国際電気通信基礎技術研究所 京都府相楽郡精華町光台二丁目2番地2 (74) 代理人 100064746 弁理士 深見 久郎 (74) 代理人 100085132 弁理士 森田 俊雄 (74) 代理人 100083703 弁理士 仲村 義平 (74) 代理人 100096781 弁理士 堀井 豊 (74) 代理人 100098316 弁理士 野田 久登 (74) 代理人 100109162 弁理士 酒井 将行</p>
---	--

最終頁に続く

(54) 【発明の名称】 マルチエージェント型制御装置およびマルチエージェント型制御プログラム

(57) 【特許請求の範囲】

【請求項1】

被制御対象に対する制御信号の生成を学習により行なうためのマルチエージェント型制御装置であって、

制御対象量を観測して獲得するための観測手段と、

観測された前記制御対象量に基づいて前記制御信号を各々算出するための、複数のエージェントの学習処理を行なうための制御処理学習手段と、

前記制御対象量が目標値に接近するに従い増加するように更新される各前記エージェントに対応する評価値と、前記複数のエージェントのリスト構造を格納するための記憶手段とを備え、

前記制御処理学習手段は、

前記リスト構造に従って、前記エージェントを順次選択しつつ、所定期間ずつ、選択された前記エージェントに対応する制御演算要素を使用して、前記制御信号を算出させ、算出された制御信号により被制御対象を駆動し、駆動後に観測された前記制御対象量に応じて、前記選択されたエージェントの評価値および前記リスト構造上、前記選択されたエージェントの所定の範囲の近傍に存在するエージェントの評価値を更新する相互作用処理手段と、

前記更新された評価値に基づいて、前記エージェントの分裂および消滅処理を行なうための分裂消滅処理手段とを含み、

前記分裂消滅処理手段は、

分裂前のエージェントの前記リスト構造に対応したリスト順に基づいて、分裂するエージェントの前または後ろに分裂したエージェントを挿入する手段と、

消滅前のエージェントの前記リスト順に基づいて、前記消滅前のエージェントの前と後ろのエージェントをつなげることで前記消滅前のエージェントを削除する手段とを含む、マルチエージェント型制御装置。

【請求項 2】

前記被制御対象に対する制御は、ブロック線図で表現され、

前記制御演算要素は、前記ブロック線図における各ブロックの前記制御信号生成のための演算処理に対応し、

各前記エージェントは、前記ブロック線図で、前記制御対象量に基づき切り替えられるモードにおいて前記制御信号を算出する複数のブロックに対応し、

前記リスト構造は、前記ブロック線図で表現される制御において前記エージェントを選択する順序に対応している、請求項 1 記載のマルチエージェント型制御装置。

10

【請求項 3】

前記制御演算要素は、前記ブロック線図におけるラプラス変換に対応する演算処理に対応し、

前記エージェントは、前記制御信号を算出するための少なくとも 1 つの組の前記制御演算要素と、モード決定要素とを有し、

前記モード決定要素は、前記制御対象量の観測値に応じて前記モードを判断し、対応する前記エージェントを発火させて前記制御信号を算出し出力させるか否かを決定する、請求項 2 記載のマルチエージェント型制御装置。

20

【請求項 4】

被制御対象に対する制御信号の生成の学習をコンピュータに実行させるためのマルチエージェント型制御プログラムであって、前記マルチエージェント型制御プログラムは、前記コンピュータに

制御対象量を観測して獲得するためのステップと、

観測された前記制御対象量に基づいて前記制御信号を各々算出するための、複数のエージェントの学習処理を行なうためのステップと、

前記制御対象量が目標値に接近するに従い増加するように更新される各前記エージェントに対応する評価値と、前記複数のエージェントのリスト構造を記憶装置に格納するステップとを実行させ、

30

前記学習処理を行なうためのステップは、

前記リスト構造に従って、前記エージェントを順次選択しつつ、所定期間ずつ、選択された前記エージェントに対応する制御演算要素を使用して、前記制御信号を算出させ、算出された制御信号により被制御対象を駆動し、駆動後に観測された前記制御対象量に応じて、前記選択されたエージェントの評価値および前記リスト構造上、前記選択されたエージェントの所定の範囲の近傍に存在するエージェントの評価値を更新するステップと、

前記更新された評価値に基づいて、前記エージェントの分裂および消滅処理を行なうステップとを含み、

前記分裂および消滅処理を行なうステップは、

40

分裂前のエージェントの前記リスト構造に対応したリスト順に基づいて、分裂するエージェントの前または後ろに分裂したエージェントを挿入するステップと、

消滅前のエージェントの前記リスト順に基づいて、前記消滅前のエージェントの前と後ろのエージェントをつなげることで前記消滅前のエージェントを削除するステップとを含む、マルチエージェント型制御プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、進化的手法によるマルチエージェントを用いた学習により制御則を最適化することが可能なマルチエージェント型制御装置およびマルチエージェント型制御プログラ

50

ムに関する。

【背景技術】

【0002】

従来、ある対象を目的の状態に制御しようとするとき、制御システムの設計が必要になる。このとき、制御対象の入出力特性が未知の場合、また、既知であっても複雑な場合、システムの設計は難しい。

【0003】

制御対象の一部が未知な場合において、制御しながら制御システムの一部を変更していく手法は、これまでも提案されている。たとえば、いわゆる「適応制御」は、制御システムを未知パラメータを用いてモデル化し、その未知パラメータを適応的に変更する。このような手法は、制御システムの特性が線形とみなせるとき有効である。

10

【0004】

さらに、ニューラルネットワークによる学習制御は、制御対象の特性が非線形な場合においても、ネットワーク構造を適応的に学習する。適切なネットワークの学習が行なわれれば、適切な制御が可能であるが、その学習は容易ではない。

【0005】

このため、非線形である制御対象は、事前にできるだけ正確にモデルし、それに基づく解析的な制御システムの設計が不可欠である。また、解析的な事前設計では制御対象モデルと実際の制御対象の間に誤差があるため、設計者による試行錯誤も重要となる場合が多い。

20

【0006】

一方、マルチエージェントにより視覚と行動を統合して制御するための制御エージェントを実現する試みについての報告がある（たとえば、非特許文献1、非特許文献2を参照）。

【0007】

この非特許文献2においては、視覚処理エージェント群の協調により特徴追跡を行い、ロボットが、任意形状ペグ挿入のタスクを行なう例が開示されている。ただし、エージェントを用いた制御についての開示はあるものの、その制御の最適化については開示がされていない。

【0008】

他方で、エージェントを用いる制御ではないものの、遺伝的アルゴリズム（GA：Genetic Algorithm）を用いて制御側を最適化する機構を利用して、非特許文献3では、並列二重倒立振り子の振り上げ安定制御を試みた例は報告されている。

30

【0009】

エージェントを利用した制御については上記のような報告例があり、また、エージェントの学習において、個々のエージェントが、システムの大域的目的やその達成度などの一部しか知覚できないため、エージェントが知覚できる範囲に基づく学習を、いかにシステム全体の大域的目的と整合させるかという問題に関しては、たとえば、エージェントが知覚できる範囲の情報である利得や報酬、評価などを個体最適性と定義し、システムの大域的目的として、すべてのエージェントがそれぞれの行動から得る利得の総和をシステム最適性とそれぞれ定義したときに、エージェント学習によるシステム最適性の向上をシミュレート可能な「動的離隔型遺伝的アルゴリズム（Dynamically Separating Genetic Algorithms、以下、DS-GAと記す）」の提案がなされている（たとえば、非特許文献4を参照）。

40

【0010】

図14は、このようなDS-GAの概念を示す図である。

このDS-GAは、個体数に応じて個体を「コロニー」と呼ぶグループに動的に離隔する遺伝的アルゴリズム（以下、GA）である。個体を離隔する遺伝的アルゴリズムとしては、他に島モデルGAなどがあるが、DS-GAは離隔状態が個体数に応じて動的に変化する点で異なる。ただし、非特許文献4に記載されるようなDS-GAについては、エー

50

ジェントをコロニーに並列に離隔することを前提としており、このような遺伝的アルゴリズムをいかにして、制御系に対して適用するかについては開示がない。

【0011】

さらに、このようなDS-GA改良して、制御システムの中にある未知パラメータをエージェントとしてモデル化し、時分割動的離隔型GA (sDS-GA) というエージェント学習手法を用いて制御パラメータの最適化を実現し、線形なロボットの制御器に応用した例も報告がある(たとえば、非特許文献5を参照)。

【非特許文献1】納谷 太、森 啓、大里 延康 著：マルチエージェント制御ロボット実験系の構築、1996年電子情報通信学会総合大会、D-8, p. 168

【非特許文献2】納谷 太、森 啓、平松 薫、大里 延康 著：多種視覚情報処理の協調に基づく能動的トラッキングとマニピュレータ行動との統合、情報処理学会第55回全国大会、Vol. 2, pp. 136-137, 1997

【非特許文献3】井口 圭一、木村 元、小林 重信：GAによる並列二重倒立振り子の振り上げ安定制御、計測自動制御学会第13回自律分散シンポジウム、pp. 277-283 (2001)

【非特許文献4】中山 功一、松井博和、野村由司彦著：動的離隔型GA (DS-GA) の提案、情報処理学会誌：数理モデル化と応用, Vol.43, No.SIG10(TOM7), pp.95-109, Nov.2002

【非特許文献5】Koichi NAKAYAMA, Hirokazu MATSUI, Katsunori SHIMOHARA, Osamu KATAI; "Proposal of Serially and Dynamically Separating Genetic Algorithm and Its Application to Optimization of Robot Control Systems," Proceedings of the Tenth International Symposium On Artificial Life and Robotics (AROB 10th), pp. GS12-2 (CD-ROM) (2005)

【発明の開示】

【発明が解決しようとする課題】

【0012】

したがって、従来は、制御対象が未知で非線形である場合においても、マルチエージェントの学習を利用して、安定する制御システムを事後適応的に設計する方法については、必ずしも明らかではない、という問題があった。

【0013】

本発明は、上記のような問題を解決するためになされたものであって、その目的は、進化的手法によるマルチエージェントを用いた学習により制御則を最適化することが可能なマルチエージェント型制御装置およびマルチエージェント型制御プログラムを提供することである。

【課題を解決するための手段】

【0014】

このような目的を達成するために、本発明のマルチエージェント型制御装置は、被制御対象に対する制御信号の生成を学習により行なうためのマルチエージェント型制御装置であって、制御対象量を観測して獲得するための観測手段と、観測された制御対象量に基づいて制御信号を各々算出するための、複数のエージェントの学習処理を行なうための制御処理学習手段と、制御対象量が目標値に接近するに従い増加するように更新される各エージェントに対応する評価値と、複数のエージェントのリスト構造を格納するための記憶手段とを備え、制御処理学習手段は、リスト構造に従って、エージェントを順次選択しつつ、所定期間ずつ、選択されたエージェントに対応する制御演算要素を使用して、制御信号を算出させ、算出された制御信号により被制御対象を駆動し、駆動後に観測された制御対象量に応じて選択されたエージェントの評価値の更新を行う相互作用処理手段と、更新された評価値に基づいて、エージェントの分裂および消滅処理を行なうための分裂消滅処理手段とを備える。

【0015】

好ましくは、分裂消滅処理手段は、分裂前のエージェントのリスト構造に対応したリス

10

20

30

40

50

ト順に基づいて、分裂するエージェントの前または後ろに分裂したエージェントを挿入する手段と、消滅前のエージェントのリスト順に基づいて、消滅前のエージェントの前と後ろのエージェントをつなげることで消滅前エージェントを削除する手段とを含む。

【 0 0 1 6 】

好ましくは、制御演算要素は、ブロック線図における演算処理に対応し、エージェントは、制御信号を算出するための制御演算要素を少なくとも1つの組有し、さらに、組の各々を活性化するか否かを制御対象量の観測値に応じて決定するモード決定要素を含む。

【 0 0 1 7 】

この発明の他の局面に従うと、被制御対象に対する制御信号の生成の学習をコンピュータに実行させるためのマルチエージェント型制御プログラムであって、プログラムは、制御対象量を観測して獲得するためのステップと、観測された制御対象量に基づいて制御信号を各々算出するための、複数のエージェントの学習処理を行なうためのステップと、制御対象量が目標値に接近するに従い増加するように更新される各エージェントに対応する評価値と、複数のエージェントのリスト構造を記憶装置に格納するステップとを備え、学習処理を行なうためのステップは、リスト構造に従って、エージェントを順次選択しつつ、所定期間ずつ、選択されたエージェントに対応する制御演算要素を使用して、制御信号を算出させ、算出された制御信号により被制御対象を駆動し、駆動後に観測された制御対象量に応じて選択されたエージェントの評価値の更新を行うステップと、更新された評価値に基づいて、エージェントの分裂および消滅処理を行なうステップとを備える。

【 発明を実施するための最良の形態 】

【 0 0 1 8 】

以下、図面を参照して本発明の実施の形態について説明する。

(本発明の概要)

以下にも説明するとおり、本発明では、制御対象の特性を同定しながら適切な制御システムを事後適応的に設計する方法を提供する。特に、本発明では、上述したような従来のDS-GAにおけるようにエージェントが時間軸上で並列に離隔できないシステムにおいて、時分割制御を用いることで、必ずしも並列処理を必要とせずにシステムへ適用可能なマルチエージェント型制御装置およびマルチエージェント型制御プログラムを提供する。

【 0 0 1 9 】

以下では、主として、本発明をいわゆる「倒立振り子」の制御に適用した場合について説明する。しかしながら、本発明は、このような構成に限定されることなく、より一般的に、ブロック線図上の複数の伝達要素の組合せにより制御則が表現されるような制御系について、適用可能なものである。

【 0 0 2 0 】

[発明の実施の形態]

(実施の形態 1)

(1 . 本発明のシステム構成)

図1は、本発明の制御方法および制御プログラムが適用される制御器を用いた制御システム1000の一例を示す概念図である。

【 0 0 2 1 】

図1を参照して、制御システム1000は、制御対象となる被制御装置200と、この被制御装置200に対して制御信号を与えるためのコンピュータ100とを備える。

【 0 0 2 2 】

図1を参照してこのコンピュータ100は、CD-ROM (Compact Disc Read-Only Memory) 上の情報を読み込むためのCD-ROMドライブ108およびフレキシブルディスク (Flexible Disk、以下FD) 116に情報を読み書きするためのFDドライブ106を備えたコンピュータ本体102と、コンピュータ本体102に接続された表示装置としてのディスプレイモニタ104と、同じくコンピュータ本体102に接続された入力装置としてのキーボード110およびマウス112とを含む。

【 0 0 2 3 】

10

20

30

40

50

図2は、このコンピュータ100のハードウェア構成をブロック図形式で示す図である。

【0024】

図2に示されるように、このコンピュータ100を構成するコンピュータ本体102は、CD-ROMドライブ108およびFDドライブ106に加えて、それぞれバスBSに接続されたCPU(Central Processing Unit)120と、ROM(Read Only Memory)およびRAM(Random Access Memory)を含むメモリ122と、直接アクセスメモリ装置、たとえば、ハードディスク124と、被制御装置200とデータの授受を行うための通信インタフェース128とを含んでいる。CD-ROMドライブ108にはCD-ROM118が装着される。FDドライブ106にはFD116が装着される。

10

【0025】

被制御装置200からは、コンピュータ100に対して被制御装置200の状態を示すパラメータ(状態量)の情報、たとえば、被制御装置200の可動部分の位置、速度、加速度、角度、角速度等の情報が与えられる。一方、コンピュータ100からは、被制御装置200に対して、これら状態量を制御するための制御情報が制御信号として与えられる。

【0026】

なお、CD-ROM118は、コンピュータ本体に対してインストールされるプログラム等の情報を記録可能な媒体であれば、他の媒体、たとえば、DVD-ROM(Digital Versatile Disc)などの他の光ディスクやメモリカードなどでもよく、その場合は、コンピュータ本体102には、これらの媒体を読取ることが可能なドライブ装置が設けられる。

20

【0027】

本発明の制御器の主要部は、コンピュータハードウェアと、CPU120により実行されるソフトウェアとにより構成される。一般的にこうしたソフトウェアはCD-ROM118、FD116等の記憶媒体に格納されて流通し、CD-ROMドライブ108またはFDドライブ106等により記憶媒体から読取られてハードディスク124に一旦格納される。または、当該装置がネットワークに接続されている場合には、ネットワーク上のサーバから一旦ハードディスク124にコピーされる。そうしてさらにハードディスク124からメモリ122中のRAMに読出されてCPU120により実行される。なお、ネットワーク接続されている場合には、ハードディスク124に格納することなくRAMに直接ロードして実行するようにしてもよい。

30

【0028】

図1および図2に示したコンピュータのハードウェア自体およびその動作原理は一般的なものである。したがって、本発明の最も本質的な部分は、FD116、CD-ROM118、ハードディスク124等の記憶媒体に記憶されたソフトウェアである。

【0029】

なお、一般的な傾向として、コンピュータのオペレーティングシステムの一部として様々なプログラムモジュールを用意しておき、アプリケーションプログラムはこれらモジュールを所定の配列で必要な時に呼び出して処理を進める方式が一般的である。そうした場合、当該制御システムを実現するためのソフトウェア自体にはそうしたモジュールは含まれず、当該コンピュータでオペレーティングシステムと協働してはじめて制御システムが実現することになる。しかし、一般的なプラットフォームを使用する限り、そうしたモジュールを含ませたソフトウェアを流通させる必要はなく、それらモジュールを含まないソフトウェア自体およびそれらソフトウェアを記録した記録媒体(およびそれらソフトウェアがネットワーク上を流通する場合のデータ信号)が実施の形態を構成すると考えることができる。

40

【0030】

[制御方法の一般的説明]

以下、本発明の構成について、その基本的な構成をまず説明する。

50

【 0 0 3 1 】

一般の制御システムは、非線形である場合も含めて入力から出力まで間の状態量をデータフローの形で表現するブロック線図で表現可能である。ブロック線図は、それぞれのブロックの入出力を他のブロックの入出力につなげることによりつくられる。これにより、個々のブロックの種類はさほど多くなくても、様々な種類の制御システムを表現することができる。

【 0 0 3 2 】

本発明では、後に説明するように、個々のブロックまたはそれらの集合をエージェントとして、ブロック線図をマルチエージェントシステムとしてモデル化する。このようにモデル化したことにより得られる制御システムのことを、以下、「マルチエージェント制御システム (Multi-Agent based Control System)」と呼ぶ。

10

【 0 0 3 3 】

すなわち、本発明では、制御パラメータのみならず、制御システムそのものをマルチエージェントシステムとしてのモデル化する。これにより、制御対象が未知で非線形な制御システムに対しても、エージェント学習手法による事後適応的な設計が実現できる。

【 0 0 3 4 】

以下の説明では、非線形制御システムの例として、モード切替え制御を取り上げ、ブロック線図をマルチエージェントシステムとして表現した例を示すとともに、その提案モデルのための新たな学習手法について述べる。すなわち、本発明においては、制御システムを構成する個々の要素 (ブロック線図におけるブロック) またはそれらの集合をエージェントとして、信号伝達などの個々の要素間の関係 (ブロック線図における線) をエージェント間の情報の入出力関係として捉え、マルチエージェントシステムによるモデル化を行なう。

20

【 0 0 3 5 】

(マルチエージェント制御システム：以下、M A C Sと呼ぶ)

上述したとおり、M A C Sでは、ブロック線図における各ブロック (伝達要素) をエージェントとして、ブロックをつなぐ線をエージェント間の関係としてモデル化する。

【 0 0 3 6 】

図3は、被制御装置200の一例である倒立振り子を示す概念図である。

ここで、倒立振り子の制御には、制御モード切替による非線形制御が必要であることが知られている。

30

【 0 0 3 7 】

図3に示すように、制御対象である倒立振り子2は、先端に質量 m の重りがついた長さ l の棒をもち、垂直二次元平面上を摩擦無しで回転できるものとする。倒立振り子2のおもりには、重力 mg が作用することになる。倒立振り子2の根元には、所定の最大出力、例えば、 $mg l / 10$ を有するモータ12と、振り子2の角度 θ を得るセンサ10が設けられている。

【 0 0 3 8 】

図4は、このような倒立振り子2を制御する制御器のブロック線図の一例である。倒立振り子2は、図4に示す制御システムにより振り上げ安定制御が可能である。

40

【 0 0 3 9 】

倒立振り子2の制御問題の目的は、適切なトルク T を出力することによって、倒立位置 (図3中の目標状態) で振子を静止させる、すなわち、基本的には、 $\dot{\theta} = 0$ の条件が満たされるような制御器を設計することである。

【 0 0 4 0 】

ここで、倒立振り子2の角度 θ がセンサ10により観測されると、制御機能ブロック302の比例伝達要素 P_t 、微分伝達要素 $D_t S$ に入力され、比例伝達要素 P_t 、微分伝達要素 $D_t S$ の出力が加え合わされて、制御機能ブロック304に与えられる。制御機能ブロック304は、制御機能ブロック302からの信号の絶対値が所定のしきい値以上であれば、スイッチ SW_1 および SW_2 をそれぞれBノード側が導通するように切換え、それ

50

以外では、スイッチSW1およびSW2をそれぞれAノード側が導通するように切換える。以後は、スイッチSW1およびSW2がそれぞれAノード側が導通するように切換えられている状態を「モードA」と呼び、スイッチSW1およびSW2がそれぞれBノード側が導通するように切換えられている状態を「モードB」と呼ぶ。

【0041】

モードAにおいては、センサ10により観測された角度 θ と値0との差が、スイッチSW1を介して、制御機能ブロック306に与えられる。制御機能ブロック306では、スイッチSW1からの信号をそれぞれ受け取る、比例伝達要素Pa、積分伝達要素Ia/S、微分伝達要素DaSの出力が加え合わされて、スイッチSW2に出力される。

【0042】

一方、モードBにおいては、センサ10により観測された角度 θ と値0との差が、スイッチSW1を介して、制御機能ブロック308に与えられる。制御機能ブロック308では、スイッチSW1からの信号をそれぞれ受け取る、比例伝達要素Pb、積分伝達要素Ib/S、微分伝達要素DbSの出力が加え合わされて、スイッチSW2に出力される。

【0043】

スイッチSW2の出力がモータ12に与えられて、モータ12の出力のトルクTにより、倒立振り子2が駆動される。

【0044】

(エージェントによる制御)

次に、エージェントによる制御の基本的なメカニズムについて述べる。

【0045】

図5は、制御エージェントの概念図である。

図5に示すように、倒立振り子2を制御するMACSは、振り子ブロックの入力であるモータトルクTを出力とし、振り子ブロック2の出力である振り子角度 θ を入力とする。各エージェントは、単位時間ごとに交代で制御を担当する。

【0046】

各エージェントは、モード決定要素Mと、図4において各ブロックで示される比例項P、積分項I、微分項Dなどの伝達要素にそれぞれ相当する「制御演算要素P, I, D」を含む。モード決定要素Mは、図4の制御機能ブロック302および304ならびにスイッチSW1およびSW2の組に対応し、モードAまたはモードBのいずれであるかを判断して、対応するエージェントを発火させる。なお、制御演算要素に対応する演算は、他の演算処理のラプラス変換に対応するものであってもよい。また、モード決定要素Mは、上述のように、モードに応じて、択一的に対応するエージェントを発火させるという処理だけでなく、たとえば、あるモードでは、対応するエージェントを発火させずに、処理を次のエージェントに渡すというような処理を行なう場合があってもよい。

【0047】

各エージェントは、外部入力(倒立振り子の角度 θ)あるいは、他のエージェント出力を入力とする。このとき、各エージェントにおいては、ブロック線図における“引き出し点”に対応して、複数の制御演算要素による互いの影響を受けない入力も可能である。入力に対して、項の特性とその係数から、各制御演算要素の出力を決定する。さらに、各エージェントにおいては、出力先を、他のエージェントあるいは、外部出力(モータトルク)とする。このとき、ブロック全図における“加え合わせ点”のように、複数の制御演算要素による出力の足し合わせも可能である。各エージェントは、上記の情報を遺伝子(制御戦略)としてそれぞれ有する。

【0048】

(MACSの制御に対するシミュレーション実験)

以下では、MACSによりモデル化した制御システムが適切に機能するか確認するために、倒立振り子2を実際に制御するシミュレーション実験について述べる。

【0049】

ここでは、MACSが所与のものである場合に、制御が適切に行なわれるかを確認する

10

20

30

40

50

ことを目的として、モード決定要素Mと制御演算要素P, I, Dの遺伝子は、それぞれ事前に求めた適切なパラメータを与える。

【0050】

図6は、シミュレーション結果を示す図である。

図6においては、実験結果の一つとして、倒立振り子の初期状態(、 d/dt)を(170、0)とし、MACSにおけるエージェント間の関係を図4で示したものと、モード数を2(A, B)、モードA(収束モード)における定数を $P_a = 10$ 、 $D_a = 10$ 、モードB(発散モード)における定数を $P_b = 0$ 、 $D_b = -0.1$ 、モード決定しきい値を $T_h = 1$ 、モード決定のための定数を $P_t = 1$ 、 $D_t = 0$ とした場合に、MACSにより制御された倒立振り子2の単位時間ごとの角度 [degree]を示す。図6により、約500単位時間までの発散モードにより振り子が頂点付近まで振り上げられた後、収束モードに切り替わり頂点へと収束していく様子が分かる。

10

【0051】

これらの結果から、MACSでは、エージェントが適切なパラメータを学習することができれば、モード切替の必要な非線形制御であっても、実現可能であることが示された。

【0052】

(エージェント学習手法)

以下では、MACSに適用するエージェント学習手法について述べる。

【0053】

非線形制御システムの評価において、現在の出力が制御システム全体の目的と合っているかどうかを即座に判断することは困難である。例えば、オーバーシュート現象のように、短期的には目標に素早く近づく出力が、長期的には目標から離れる場合もある。そのため、MACSにおいて、それぞれのエージェントが担当する短時間の出力が、長期的に適切な出力であるかどうかは即座には判断できない。

20

【0054】

本発明が適用可能な制御対象である倒立振り子2は、不安定であり、一旦、不安定になると、適正な制御システムに変更しても、系が安定しない特性がある。すなわち、このような制御はいわゆる「ジレンマ環境」において、安定収束性の高いシステム最適な解を得ることに相当する。

【0055】

図7は、図1および図2に示した制御システム1000の構成を示す機能ブロック図である。

30

【0056】

センサ10は、倒立振り子2の角度を計測(観測)し、制御部100のセンサ系インタフェース128aに観測データを与える。

【0057】

制御部100では、センサ系インタフェース128aを介して入力された観測データは、一旦、データバッファ部として機能するメモリ122に格納され、制御処理学習部として機能するCPU(Central Processing Unit)120が、メモリ122中の観測データを用いて、たとえば、所定時間単位ごとに、倒立振り子2の倒立動作を制御するための処理を実行する。

40

【0058】

すなわち、CPU120では、記憶装置として機能するハードディスク124に格納されているプログラムに基づいて、観測データに基づいて、駆動モータ12を制御するための制御信号、たとえば、駆動モータ12が出力する駆動トルクTに対応した制御信号を計算して、駆動系インタフェース128bに与える。

【0059】

(リスト構造型GA)

次に、制御処理学習部が行なう学習処理について、さらに詳しく説明する。

【0060】

50

図8は、リスト構造型遺伝アルゴリズム（リスト構造型GA）における遺伝子個体（エージェント）の分裂処理、および消滅処理を説明するための図である。

【0061】

リスト構造型GAは、遺伝子個体はリスト構造により表現され、一列に並んだ遺伝子個体の相互作用を隣接個体間（または近傍個体間）に制限する。遺伝子個体は分裂と消滅により増減する。

【0062】

個体の評価値が初期値の、たとえば、倍以上になると個体は分裂し、新たに生まれた個体が分裂前の個体の隣（当該個体の前または後。事前にいずれにするか設定しておくか、あるいは、ランダムに選択する。）に挿入される（図8（A））。このとき、無作為に選択された個体が消滅する。

10

【0063】

一方、個体の評価値が0以下になると個体は消滅し、消滅した個体に隣接していた個体同士が新たに隣接個体となる（図5（B））。このとき、無作為に選択された個体が分裂する。

【0064】

リスト構造型GA（LS-GA）は、DS-GAと同様に選択オペレーション以外の個体間相互作用を制限することにより、個々の遺伝子個体にとってのみ適切な出力ではなく、集団全体にとって適切な出力を獲得する手法である。

【0065】

図9は、メモリ122（または必要に応じてハードディスク124）に格納される、エージェントのリスト構造を説明するための図である。

20

【0066】

図9（a）に示すように、このテーブルの横一列が、単位時間ごとに処理するエージェントである。各エージェントは、評価値 $E(a_i)$ と、制御における入出力関係を示す（制御システムの構造を含む）制御戦略 $S_t(a_i)$ を有する。

【0067】

図9（a）において、ポインタ（データの格納場所）とは、メモリ122等における記憶領域内の物理的な位置を示す。一方、各エージェントに対しては、自身の直前のエージェントのポインタと、自身の直後のエージェントのポインタもテーブル中に格納されている。このようなポインタにより接続関係を表現することで、エージェントのリスト構造を実現できる。

30

【0068】

i) 後に説明するように、単位時間ごとに自身の制御に伴う観測データ（現在の制御対象の状態）により、現在の制御を担当しているエージェントの評価値を決定が決定され、このエージェントの評価値が増減する。

【0069】

ii) さらに、単位時間経過後に、現在、制御を担当しているエージェントの次エージェントのポインタに存在するエージェントに制御担当が交代する。このi)およびii)の処理の繰り返しにより、エージェントは図9（b）のように順番に制御を担当する。

40

【0070】

図10は、各エージェントの制御戦略 $S_t(a_i)$ の例を示す図である。

図10に示すように、各エージェントについては、制御戦略 $S_t(a_i)$ は、入力データの入力元を特定するための入力元情報、入力元から入力された入力データ、出力データの出力先を特定するための出力先情報、出力される出力データの属性データを含む。

【0071】

制御戦略 $S_t(a_i)$ は、さらに、エージェントを構成する制御演算要素の種類を特定するための制御演算要素情報、制御演算要素が行なう演算処理を特定するための演算パラメータの属性データを含む。さらに、制御戦略 $S_t(a_i)$ は、付随属性として、制御演算要素の入力の態様（たとえば、引き出し線に対応するの否か、あるいは、入力元から

50

のデータに共通に所定の演算処理を行なうのか否か)を特定する情報や、制御演算要素の出力の態様(たとえば、加え合わせ点に対応するのか否か等)を特定する情報を含む。

【0072】

図9(a)および図10に示す情報が、各エージェントの遺伝情報となる。

図11は、制御処理学習部が行なう学習処理を説明するためのフローチャートである。

【0073】

以下、図11を参照して、LS-GAのアルゴリズムについて、さらに詳しく説明する。

【0074】

まず、以下のような初期設定の処理が行われる(ステップS100)。

10

(1)初期設定

すなわち、制御処理学習部120は、LS-GAを適用するN個(N:2以上の自然数)の制御エージェントをランダムに生成する。なお、エージェントの初期個数Nは、CPU120の処理能力を考慮して、その個数を予め設定するものとする。

【0075】

ここで、駆動トルクTを計算するための制御演算要素の種類と演算パラメータ、制御演算要素の入力の態様、制御演算要素の出力の態様は、特に、限定されないが、予め定められた範囲内の選択肢からランダムに選ばれるものとする。

【0076】

このような制御エージェントのランダムな生成を「エージェントの初期生成」と呼ぶことにする。

20

【0077】

さらに、各エージェントごとに、評価値E(ai)の値が初期値として、たとえば、0に設定される。

【0078】

再び、図11を参照して、次に、学習処理中のエージェントの最大数Nagentを、特に限定されないが、たとえば、Nに設定する(ステップS102)。

【0079】

さらに、初期設定として、時間を表す変数t1の値は0とされる(ステップS104)。

30

【0080】

(2)相互作用:

続いて、各エージェントは、以下に説明するような行動と相互作用を行なう。

【0081】

まず、リスト構造に従い、制御を担当するエージェントを選択する(ステップS106)。

【0082】

続いて、選択されたエージェントの制御戦略St(ai)に基づいて、制御信号が計算される(ステップS108)。

【0083】

続いて、計算された制御信号により、所定時間tcの間だけ、駆動モータ12が駆動され、対応するトルクTが倒立振り子2に与えられる(ステップS110)。

40

【0084】

さらに、ステップS110の駆動の間に、倒立振り子2の角度θがセンサ10により観測される。このとき、倒立振り子2の角度θおよび角度θの時間微分であるdθ/dtにより、たとえば、以下の式に基づいて、選択されて制御を担当した制御エージェントの評価値が更新される(ステップS112)。

【0085】

$$E(a_i) = E(a_i) + (E_0 - | \quad |) + (E_1 - | d \theta / d t |) \dots (1)$$

ここで、E0、E1は所定の定数である。また、|...|は絶対値を表す。

50

【 0 0 8 6 】

さらに、選択されて制御を担当した制御エージェントにリスト構造上隣接する（または、リスト構造上、所定の範囲の近傍に存在する）エージェントの評価値も式（1）により更新される。このとき、制御担当のエージェントからの距離に応じて、所定の係数で評価の更新分を減衰させてもよい。

【 0 0 8 7 】

なお、評価値は、ステップ S 1 1 0 の駆動の終了時点で、測定された角度 と、それに対応する時間微分により計算されるものとする。

【 0 0 8 8 】

また、評価値の計算は、上記式（1）の方法に限定されることなく、より目標に近づくほど、順次、より評価値が増大するような関数であれば、他の計算式によってもよい。

10

【 0 0 8 9 】

（3）エージェントの分裂と消滅：

エージェントは、エージェントの評価値が所定の値、たとえば、初期値の倍以上になると分裂を起こすものとする。このとき、各エージェントは、分裂前の評価値を半分ずつ持つ2個体に分裂し、それぞれ遺伝子を引き継ぐ（ステップ S 1 1 4）。このとき、無作為に選択された個体が消滅する。

【 0 0 9 0 】

また、エージェントは、自身の評価値が0以下になると消滅するものとする（ステップ S 1 1 6）。このとき、無作為に選択された個体が分裂する。

20

【 0 0 9 1 】

なお、遺伝子が、所定の突然変異確率 p_1 で変異するものとしてもよい。すなわち、たとえば、ランダムに選択されたエージェントについて、この突然変異確率 p_1 で、再度、「エージェントの初期生成」処理により、遺伝子が変異することとしてもよい。

【 0 0 9 2 】

次に、時間 t が、 $t = t + t_c$ と更新され（ステップ S 1 1 8）、続いて、制御処理が終了したか、たとえば、時間 t が、所定の処理時間 T_0 よりも大きいか判断される（ステップ S 1 2 0）。

【 0 0 9 3 】

処理が終了していなければ、処理は、ステップ S 1 0 6 に復帰する。

30

以上のステップ S 1 0 6 ~ ステップ S 1 2 0 での処理を1単位時間とし、CPU 1 2 0 は、この処理を繰り返すことで、エージェントの学習処理を実行する。

【 0 0 9 4 】

なお、上述した説明では、ステップ S 1 1 4 において、無作為な個体の消滅が起こり、ステップ S 1 1 6 においては、無作為な分裂が起こるものとした。これは、本発明の方法では、必ずしもエージェントの総数に制限はないものの、計算のコストを一定にする観点から、エージェントの最大数を一定にするために行なわれるものである。このとき、上述した説明では、エージェントの総数は、初期値から変化しないことになる。しかし、たとえば、ステップ S 1 0 2 において、エージェントの最大数 N_{agent} を、初期値の N ではなく、他の値に設定しているのであれば、ステップ S 1 1 4 においては、たとえば、エージェントの総数が N_{agent} を超えるときは、総数が N_{agent} となるまで、ランダムにエージェントの消滅処理を行ない、ステップ S 1 1 6 においては、ランダムに個体を選択して分裂させる処理を、エージェントの総数が N_{agent} となるまで、繰り返す、こととしてもよい。

40

【 0 0 9 5 】

以上のような処理においては、時間的にエージェントが入れ替わるので、実際には、過去に担当したエージェントの行動により、現在のエージェントの評価が変わる。したがって、一般には、現在のエージェント単体の評価がしにくい。

【 0 0 9 6 】

しかしながら、上述したような集団最適性を得る性質を持つ LS - GA を用いれば、個

50

々に評価されるエージェントを、集団最適性をもつように学習させることができる。

【0097】

以上のような処理により、エージェント間の影響を近接したエージェントに限定するLS-GAによって、仮想的な離隔を実現して、エージェントの処理を並列に行なうことができないハードウェアの制御に対して、制御則の最適化を自律的に適応的に行なうことが可能となる。

【0098】

(LS-GAとDS-GAの比較)

以下では、LS-GAとDS-GAをジレンマモデルに適用した場合の比較結果を、両者のジレンマ環境への適用可能性を比較するものとして説明する。

10

【0099】

(囚人のジレンマモデルによる比較実験)

図12は、マルチエージェントの標準問題である囚人のジレンマモデルを示す図である。

【0100】

以下では、従来のDS-GAとLS-GAを図12に示すような囚人のジレンマモデルに適用し、それぞれの性質を比較する。

【0101】

囚人のジレンマモデルに対する個体間相互作用を以下に示す。

(1) 単位時間ごとに、LS-GAでは隣接個体と、DS-GAでは同一コロニー内から無作為に選択した個体と対戦する。

20

【0102】

(2) 対戦する2個体は、裏切Dを選択する確率を示すそれぞれの遺伝子GeneAct(a) (0/10, 1/10, ..., 10/10) に従い、協調、または裏切を選択する。

【0103】

(3) 互いの選択行動の組合せに応じて利得を得る。

この実験では、初期個体数10000、DS-GAの限界数10、突然変異確率0.001、初期評価値20とし、移動は用いない。また、異なるランダム系列を用いて100試行する。

【0104】

30

図13は、実験結果の一つとして、遺伝子別個体数比率の推移を示す図である。

図13を参照すると、LS-GAおよびDS-GAのいずれの学習手法を用いた場合においても、集団全体にとって適切な協調行動を学習したことが分かる。特に、DS-GAの場合には最初に裏切個体が増加し、その後協調個体の増加に転じるのに対し、LS-GAの場合には裏切個体が増加せずに、すぐに協調個体が増加した。

【0105】

これらの結果から、エージェント学習にリスト構造型GAを適用した場合には、動的離隔型GAを適用した場合と同様に、ジレンマ環境においても集団全体にとって適切な結果を学習することが示された。

【0106】

40

この実験で示されたように、LS-GAは、DS-GAと同様に全体最適な出力を学習できる。すなわち、DS-GAの適用が容易な対象はDS-GAにより学習し、LS-GAの適用が容易な対象はLS-GAにより学習することが可能である。

【0107】

今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【図面の簡単な説明】

【0108】

50

【図1】本発明の制御方法および制御プログラムが適用される制御器を用いた制御システム1000の一例を示す概念図である。

【図2】コンピュータ100のハードウェア構成をブロック図形式で示す図である。

【図3】被制御装置200の一例である倒立振り子を示す概念図である。

【図4】倒立振り子2を制御する制御器のブロック線図の一例である。

【図5】制御エージェントの概念図である。

【図6】シミュレーション結果を示す図である。

【図7】図1および図2に示した制御システム1000の構成を示す機能ブロック図である。

【図8】リスト構造型GAにおける遺伝子個体(エージェント)の分裂処理、および消滅処理を説明するための図である。

10

【図9】メモリ122に格納される、エージェントのリスト構造を説明するための図である。

【図10】各エージェントの制御戦略St(ai)の例を示す図である。

【図11】制御処理学習部が行なう学習処理を説明するためのフローチャートである。

【図12】マルチエージェントの標準問題である囚人のジレンマモデルを示す図である。

【図13】遺伝子別個体数比率の推移を示す図である。

【図14】DS-GAの概念を示す図である。

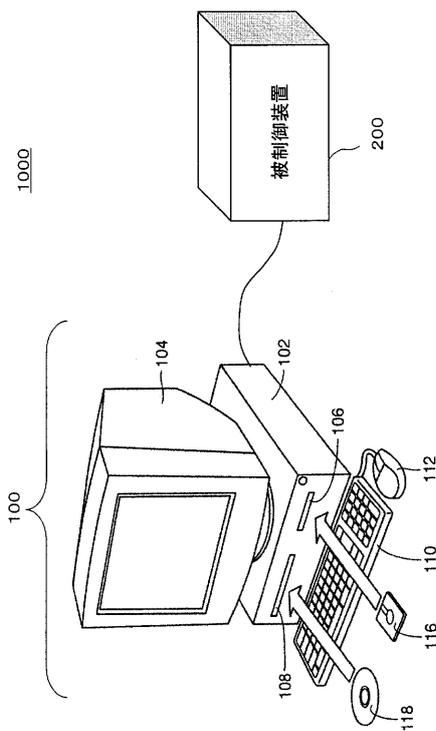
【符号の説明】

【0109】

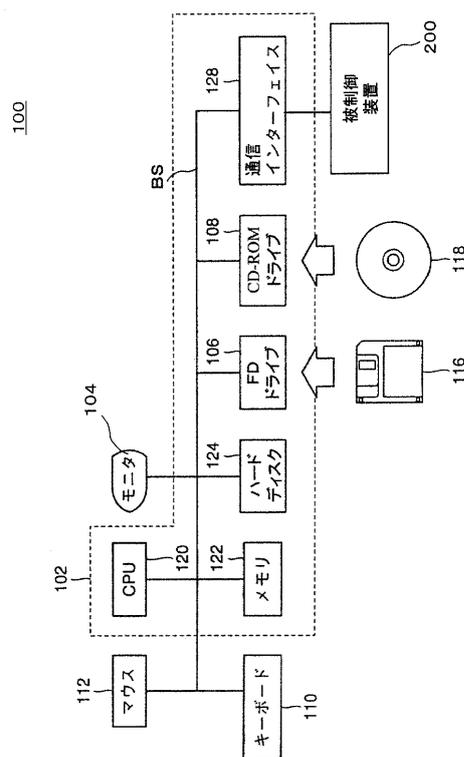
20

2 倒立振り子、10 センサ、12 駆動モータ、100 制御部、122 データバッファ部、124 記憶装置、120 制御処理学習部、128a センサ系インタフェース、128b 駆動系インタフェース、200 被制御装置、1000 制御システム。

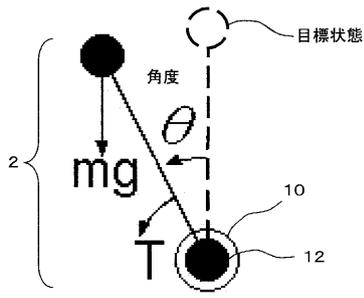
【図1】



【図2】

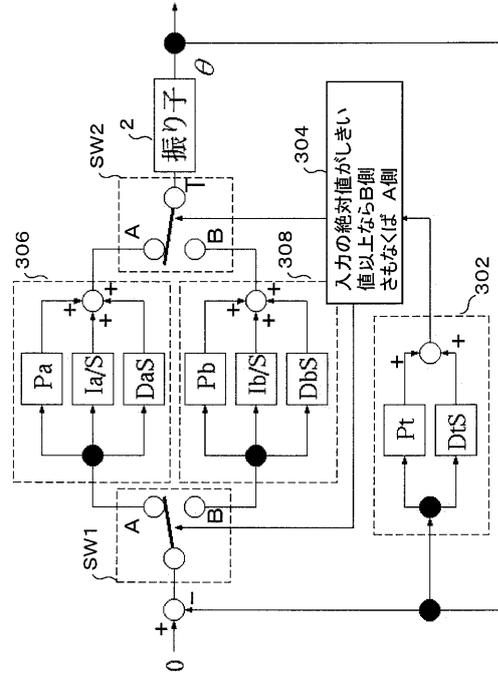


【図3】

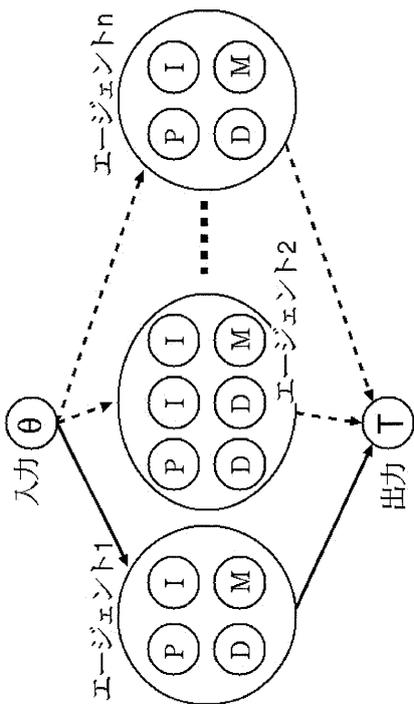


200

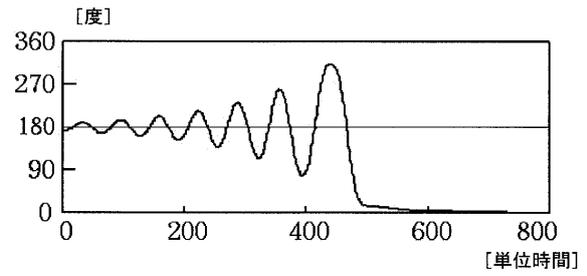
【図4】



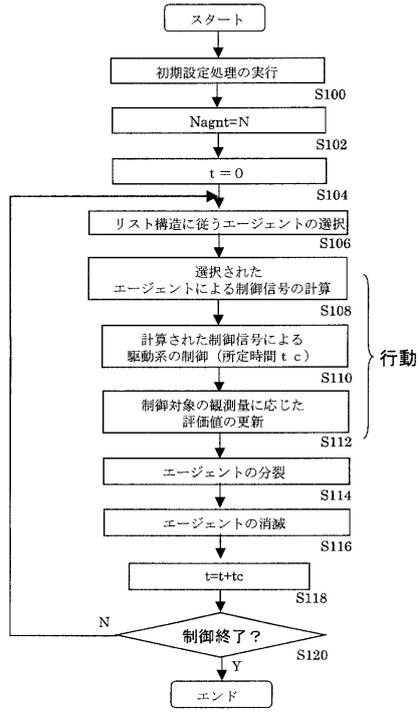
【図5】



【図6】



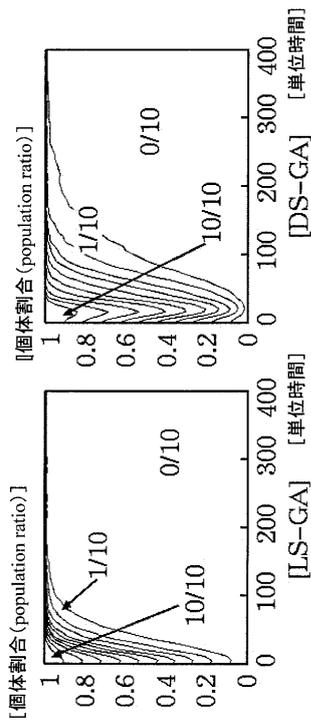
【図 1 1】



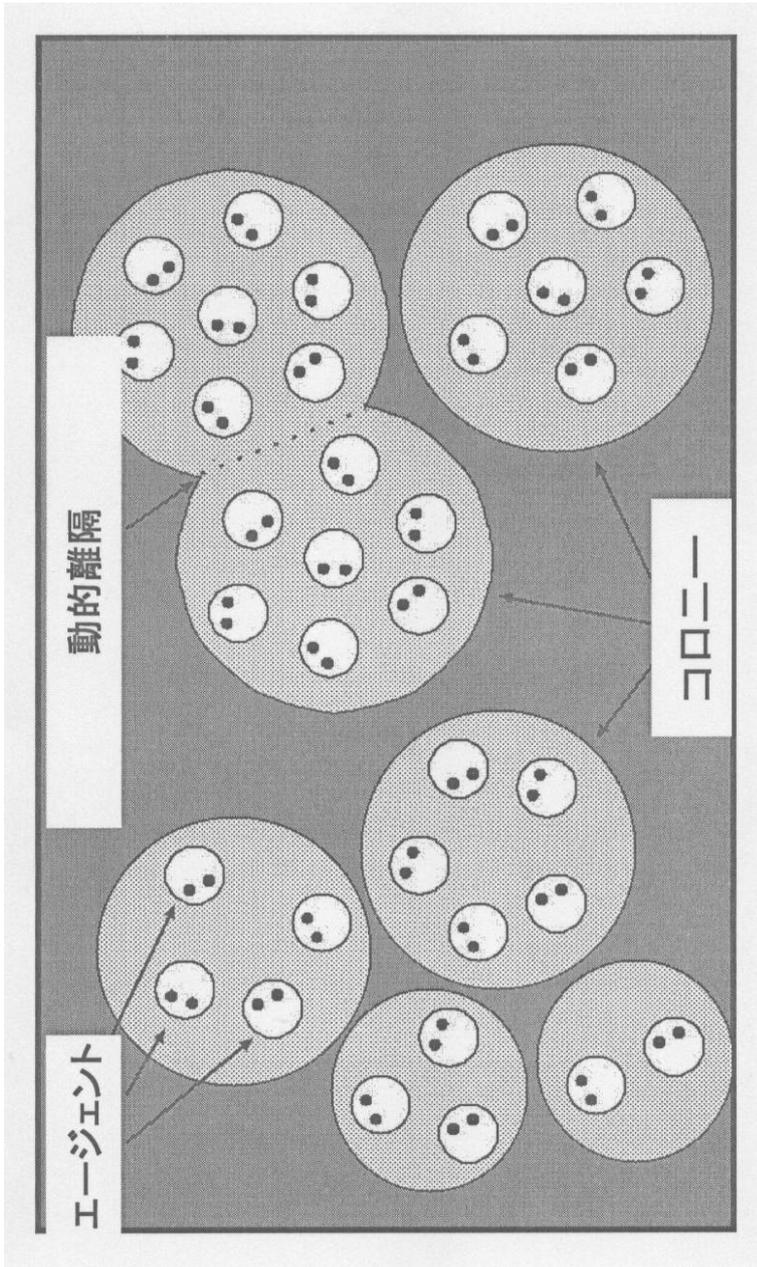
【図 1 2】

自分 \ 相手	協調C	裏切D
協調C	4	1
裏切D	5	2

【図 1 3】



【図14】



フロントページの続き

- (72)発明者 中山 功一
京都府相楽郡精華町光台二丁目2番地2 株式会社国際電気通信基礎技術研究所内
- (72)発明者 下原 勝憲
京都府相楽郡精華町光台二丁目2番地2 株式会社国際電気通信基礎技術研究所内

審査官 長谷川 篤男

- (56)参考文献 中山 功一, 要素間相互作用の動的離隔を内包した創発的システム設計法の構築, 2005年3月, 第167-171頁, URL, <http://www.fu.is.saga-u.ac.jp/~knakayama/introduction.html>
- 中山 功一 他, 階層化動的離隔型GA(hDS-GA)による離隔パラメータの最適化, 情報処理学会論文誌, 2004年2月15日, 第45巻 No. SIG2(TOM10), 第42-55頁
- 中山 功一 他, DS-LAによるコンピュータの創発的ネットワーク生成に向けた研究, システム制御情報学会研究発表講演会講演論文集, 2004年, 第617-618頁

- (58)調査した分野(Int.Cl., DB名)
G06N 3/00
JSTPlus(JDreamII)