コンセンサスな制御はいかが?

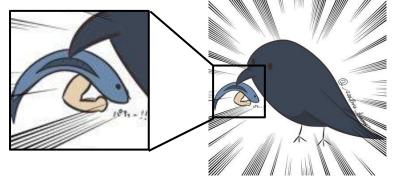
~合意制御についてちょっと触れてみよう~

ざきまつ

April 6, 2024

はじめに

- 初めまして、「ざきまつ」です。
- カラスに咥えられている魚が本体です
- ツイ廃です
- 大学院新入生です



X @santana_hammer

合意制御ってなんなのさ?

皆様、「合意制御」という言葉を聞いたことありますか?

聞いたことある方:ここから先は寝ていただいても大丈夫です. おやスヤァ..

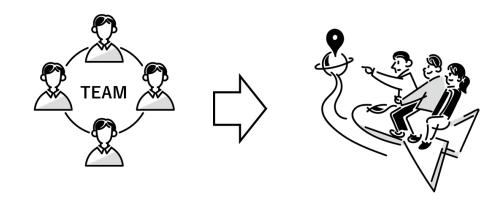
聞いたことない方:僕も研究室に配属されるまで聞いたことありませんでした.

仲間です,仲良くしてください.

マルチエージェントシステム (MAS)

複数の自律的なエージェントが相互作用し、目標を達成するために協力するシステム

- エージェント: 自律的に動作する個体(人間、ロボット、ソフトウェアなど)
- 自律: 外部からの直接的な介入なく, 自分自身の判断で行動を選択
- 相互作用: エージェント同士が情報を共有したり, 行動を調整したりするプロセス



マルチエージェントシステム (MAS)

どんな特徴があるのか?

- ◆ 分散性: システム全体の制御が集中型ではなく, 各エージェントに分散している。
- スケーラビリティ: エージェントを追加することで, システムの規模を柔軟に調整可能
- ロバスト性: 一部のエージェントが失敗しても、システム全体としては機能を維持可能
- 適応性: 外部環境の変化に対して、エージェントが自律的に行動を調整可能

合意制御ってなんなのさ?

合意制御(Consensus Control)

複数のエージェント(ロボット、センサー、ネットワーク上のノードなど)が情報を共有し合い、共通の目標に到達するための手法

目的

- 複数のエージェントが、互いの状態(位置、速度、方向など)について合意を達成
- エージェント間で共通の目標値や動作を実現

合意制御ってなんなのさ?



合意制御の数式

合意アルゴリズム

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (x_i(t) - x_j(t))$$

where,

- $u_i(t)$ is the control input for agent i at time t.
- \mathcal{N}_i denotes the set of neighbors of agent i.
- $x_i(t)$ and $x_j(t)$ represent the states of agent i and its neighbor j at time t, respectively.
- 通信可能(情報交換・参照が可能)なエージェントとの状態の差を計算
- 計算した差分を元に制御入力を決定
- いい塩梅を探しながら一つの状態に収束していく。

合意制御の数式

少し式変形

$$\begin{bmatrix} \dot{x}_{1}(t) \\ \dot{x}_{2}(t) \\ \vdots \\ \dot{x}_{n}(t) \end{bmatrix} = \begin{bmatrix} -\sum_{j=1}^{n} a_{1j} (x_{1}(t) - x_{j}(t)) \\ -\sum_{j=1}^{n} a_{2j} (x_{2}(t) - x_{j}(t)) \\ \vdots \\ -\sum_{j=1}^{n} a_{nj} (x_{n}(t) - x_{j}(t)) \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{j=1}^{n} a_{1j} & -a_{12} & \dots & -a_{1n} \\ -a_{21} & \sum_{j=1}^{n} a_{2j} & \ddots & -a_{2n} \\ \vdots & \ddots & \ddots & -a_{(n-1)n} \\ -a_{na} & \dots & -a_{n(n-1)} & \sum_{j=1}^{n} a_{nj} \end{bmatrix} \begin{bmatrix} x_{1}(t) \\ x_{2}(t) \\ \vdots \\ x_{n}(t) \end{bmatrix}$$

$$\dot{x}(t) = -Lx(t)$$

ちょっとグラフ理論

グラフラプラシアン (Graph Laplacian)

- グラフの構造を表す行列で、次数行列と隣接行列の差によって定義される
- エージェント間の相対的な位置関係や、全体としての接続性の特性を分析するのに使用される。

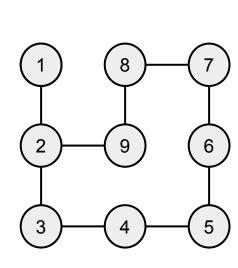
隣接行列 (Adjacency Matrix)

- エージェント(またはノード)間の接続を示す行列
- 行と列はシステム内のエージェントを表し、各要素の値はそのエージェント間の接続を表す (通常、接続がある場合は 1、ない場合は 0)

次数行列 (Degree Matrix)

- 各エージェント(ノード)がどれだけ多くのエージェントと接続しているかを示す対角行列
- 対角要素はそのノードの「次数」(接続の数)を示し、非対角要素は全て 0

ちょっとした例



$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \\ \dot{x}_6(t) \\ \dot{x}_7(t) \\ \dot{x}_8(t) \\ \dot{x}_9(t) \end{bmatrix} = -\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_7(t) \\ x_8(t) \\ x_9(t) \end{bmatrix}$$

各時間系

連続時間バージョンと離散時間バージョンがある(当たり前)

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (x_i(t) - x_j(t))$$

離散時関系

$$u_i[k] = -\varepsilon \sum_{j \in \mathcal{N}_i} (x_i[k] - x_j[k])$$

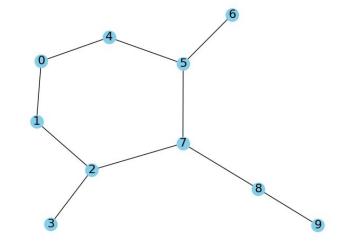
離散時関系については、ペロン行列と呼ばれる行列を用いて、簡潔に表現できる。

$$x[k+1] = x[k] + u[k]$$
$$= Px[k]$$

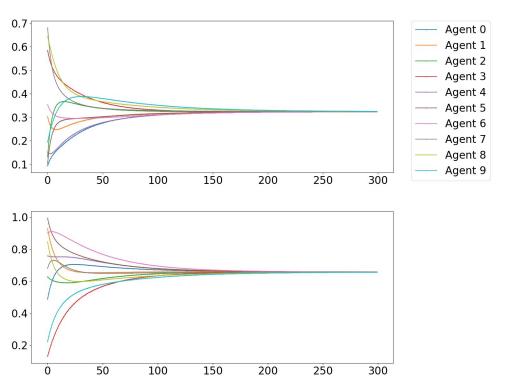
where $P = I - \varepsilon L$ and $\varepsilon < 1/\Delta$.

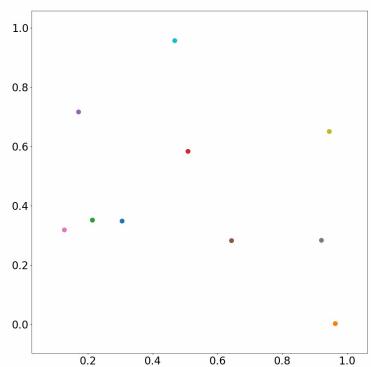
シミュレーション例

$$n = 10, d = 2, \varepsilon = 0.05, k \in [0, 300]$$

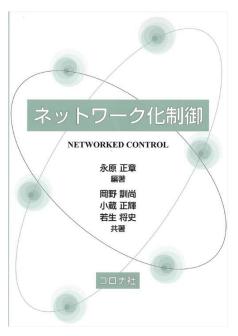


シミュレーション例





システム制御工学シリーズ 22 マルチエージェントシステムの 博士(情報学) 永原 正章 Ph.D. 石井 秀明 CO 博士(工学) 林 博士(情報学) 桜間 一徳 博士(情報学) 畑中 健志 コロナ社



コロナ社

Institute of Systems, Control and Information Engineer

システム/制御/情報, Vol. 57, No. 5, pp. 211-218, 2013

講座

1. HI: M.C.

マルチエージェントシステムの制御 — T 総論

石井 秀明*

ングの観点から見た後、6. ではシステム全体でのネット ワーク構造の表現法について概説する。7.では、電力シ

多数の自律的に意思決定を行うことのできる構成要素

から成るシステムをマルチエージェントシステムとよぶ。 各要素をエージェントとよぶが、それらが相互に影響を 及ぼしあうことで、システム全体のレベルでの振舞いが 定まる、とくに工学的なシステムにおいては、エージェ ント間で共通の目標を達成することが目的となる。

近年のセンサやアクチュエータ技術の向上により、小 型機器であっても無線通信を行ったり、一定の計算を行 うことができるものが比較的安価に手に入るようになっ た。制御システムにおいてもネットワーク化が可能とな り、大規模・複雑化してきた結果、そのようなシステム をどのように協調制御すべきかが、制御工学の観点から も新たな研究対象となってきた。そこでは特定の制御タ スクを実行したいときに、どのような相互作用・情報交 株 あるいは分数型制御アルゴリズムを実行すれば達成 が可能であるかが重要な課題となっている。

過去10年ほどの間、こうした課題は制御工学の分野 で多くの研究者の関心を集め、熱心に研究がなされて きた (1.6.9.23)。その理由の一つに関連する応用分野が 非中に傾けいことがあげられる 従来 制御工学が終っ てきたアプリケーションに近いものでは、センサネット ワークや自律移動型群ロボットがある。より最近は、電 カネットワーク、システムバイオロジー、あるいは社会 的ネットワークなどの関連する分野に深く関わる形で発 MI THA

本講座の目的は、前回 例でも述べられたとおり、シス テム制御の観点からこうしたマルチエージェントシステ ムの制御に関する初等的な内容を体系立てて紹介するこ とである。今回はその導入部にあたり、マルチエージェ ントシステムの代表的な応用や関連する課題についてま

本稿の構成は以下となる。まず2.では、動物の群れ行 動をモデル化したBoids とよばれるシミュレーションに ついて説明する、つぎに3.では、自律移動型ロボットに 類様する応用をいくつか紹介する。4. ではセンサネット ワーケにおける通信や禁護器について述べる。5. でエー ジェント基での相互作用の立ち方について通信とセンシ * 東京工業大学 大学院 総合理工学研究科

distributed control systems

ステムや情報科学などにおけるエージェント系として扱 うことのできるシステムや課題を紹介する、最後に8.で

2. 動物の協調行動モデル: Boids

自然界における協調制御の代表的な例として、魚や 鳥の群れ行動がある。そこでは多数の個体が、集まる まとまって移動する・フォーメーションを組むなど、社 会的を集団行動を取ることができる。こうした現象は 広くマルチエージェントシステムの研究を行う際の動機 づけとなり、目指すべき目標とされてきた、個々の動物 (ロエージェント) がどのような意思決定を行っている かを考えるうえで、米国の研究者 Reynolds によるコン ピューケグラフィクスの Boids はよくしられ、多くの示 唆を与えてきた[27]。本前では、その概要を説明する。 資料のエージェントが一切速度で放動! ている対況を

老える。 進行方向については、おのおののエージェント は、位置的に近い他のエージェントの行動を考慮して決 定するとしよう、どのようなルールに基づいて進行方向 を決めれば、すべてのエージェントが同一の方向に向く ような協調行動を示すことができるだろうか?また、考 速すべき他のエージェントについては、魚や鳥であれば 視覚で感知できる範囲内にいる仲間程度であろうから、 第1国に示すように、一定のセンサレンジ内にいるもの

無論、グループに指揮を執るリーダがいれば、エー ジェント系のモデリングは比較的簡単である。しかし、 ここではリーダの存在は仮定せず、全エージェントが対 等で等しいルールに従って行動するものとしよう。さら . ルールといっても (動物であるので) 可能な限り単 絶なものを想定したい、Boids のモデルでは、以下の三 つのルールを採用した(第2回).

- 衝突回避:エージェント同士が衝突しないよう。混 誰してきたら離れる方向に向きを求える。
- 整列:レンジ内のエージェントの進行方向の平均に あたる方向を向く。
- 結合:レンジ内のエージェントの位置の重心にあた る場所に進行方向を向ける。 上の三つのルールはいずれも、各エージェントが自身

Institute of Systems, Control and Information Engineers

システム/制御/情報、Vol. 57, No. 7, pp. 283-292, 2013



マルチエージェントシステムの制御 - II 代数的グラフ理論

林 南樹* · 永阪 正意 †

講座

1. はじめに マルチエージェントシステムの協調制御では、相互作

用(または情報伝達)をとおしたエージェント間の「つ

ながり方しの構造が入わめて重要になる。そのような

「つながり方」を表す数学モデルとして、グラフ (graph)

グラフを扱う理論はグラフ理論とよばれる。その中でも

特に、線形代数を用いてグラフの性質を解明する代数的グ

ラフ理論 (Algebraic Graph Theory) [2.9.18] は、マル

チェージェントシステムの制御における解析・設計の手法

としてきわめて効力をツーもである [1 3 8 13 14 16 17]

本連察では、マルチエージェントシステムの制御を学

ぶうえで基礎となる代数的グラフ理論を概認する。まず

代数的グラフ理論を学ぶために必要な親形代数の知識。

然に大学の学郎ではあまり寄うことのない事実につい

て述べ つぎに作数的グラフ要論について マルチエー

ジェントシステムの制御に必要な項目に焦点を絞って解

本能では、代数的グラフ理論で領額に用いる線形代数

の知識を概載する。定理の証明はすべて省略するが、それ

らは、線形代数の標準的な教料書、たとえば [10,12,15]

ここでは準備として、本講座で用いるおもな数学記号

実数、整数、自然数、複素数の集合をそれぞれ限、Z、

N.C で表し、非負実数および非負額数の集合をそれぞ

れ及、および区、で表す、また、実部が非負である複素

数の集合 (関右半平面) を €, で表す、複素数 z ∈ € に

対して、|z| はその絶対値、 $\angle z$ はその偏角、 $\mathrm{Re}\,z$ およ

びIm:はそれぞれ:の実部および虚部、ミは:の模素

集合 X の元を要素にもつm×n行列の集合を X^{m×n}

で表し、n=1のとき〈すなわち m 次元縦ベクトルの集

および用語の定義と行列に関するいくつかの基本性質に

御本行う

を参照されたい。

2.1 準備

ついて述べる

ま行を表す。

* 大阪大学 大学院 工学研究科

algebraic graph theory.

1 京都大学 大学院 情報学研究科

2. 線形代数の基礎

合) は \mathcal{X}^m と表記する。たとえば、 $\{0,1\}^{m \times n}$ は0また は1を要素にもつ $m \times n$ 行列の集合である。第 (i,j)要 素が a_{ij} である行列を $[a_{ij}]$ と略記する。

行列 $A = [a_{ij}]$ に対して、 A^{\top} で転置を、 A^{\bullet} で複素共役 転置を表す、すなわち、 $A^{T} = [a_{ji}], A^{*} = [a_{ji}]$ である。サ イズがnの単位行列を In、要素がすべて1のn 次元級ベク トルを 1 と表す (派え字 n はサイズが明らかなときは名 略する)、実数 (または複素数) $x_1, x_2, ..., x_n$ を対角要素に もつ対角行列を $\mathrm{diag}\{x_1,x_2,...,x_n\}$ 、行列 $A_1,A_2,...,A_n$

を対象プロックにもつプロック対象行列(または 行 列 $A_1,A_2,...,A_n$ の直和)を blkdiag $\{A_1,A_2,...,A_n\}$ で 二つのベクトル $x,y \in \mathbb{R}^n$ の標準内積 (standard inner product) または Euclid 内積 (Euclidean inner product) を $\langle x,y \rangle := y^{\top} x$ で定義し、ベクトル x の Euclid J

ルム (Euclidean norm) を $|x| := \sqrt{(x,x)}$ で定義する。 また、実数 $p \ge 1$ に対して、ベクトル $x = [x_1,...,x_n]$ のp-ノルム (p-norm) を

 $||x||_p := \left(\sum_{i=1}^n |x_i|^p\right)$

で定義する、ベクトル $x,y \in \mathbb{R}^n$ に対して、 $\langle x,y \rangle = 0$ が 成り立つとき、x と y は直交する (orthogonal) といい。 エムタと表記する. 行列 $A \in \mathbb{R}^{m \times n}$ の零化空間またはカーネル (bernel) を

 $\ker(A) := \{x \in \mathbb{R}^n : Ax = 0\}$

像空間またはイメージ (image) を $\operatorname{imag}(A) := \{y \in \mathbb{R}^m : y = Ax, \ x \in \mathbb{R}^n\}$

で定義する。像空間 imag(A) の次元を行列 A の閲数 (rank) とよび、rank(A) で表す。行列 A の零化空間 $\ker(A)$ と像空間 $\operatorname{imag}(A)$ の次元について、次の定理が

【定理 1】 (次元定理) 行列 A c R **** に対して $\dim \ker(A) + \operatorname{rank}(A) = n$

- 21

正方行列 A の行列式を det(A) で表す。二つの正方行

利 A と A' が相似 (similar) であるとは、ある正明行列

NII-Electronic Library Service

NII-Electronic Library Service

システム/制御/情報

マルチエージェントシステムの制御

コロナ社 マルチエージェントシステムの制御

ネットワーク化制御