社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

[招待講演] 自己組織型ネットワークシステムの適応性向上に関する研究

小南　大智†　加嶋　健司††　橋本　智昭†††　村田　正幸††††

† 大阪大学 大学院経済学研究科 〒 560-0043 豊中市待兼山町 1-7
†††† 大阪大学 大学院情報科学研究科 〒 565-0871 吹田市山田丘 1-5
††† 大阪大学 大学院基礎工学研究科 〒 560-8531 豊中市待兼山町 1-3
†† 京都大学 大学院情報学研究科 〒 606-8501 京都市左京区吉田本町 36-1
E-mail: †d-kominami@econ.osaka-u.ac.jp, ††kashima@amp.i.kyoto-u.ac.jp, †††thashi@sys.es.osaka-u.ac.jp,
††††murata@ist.osaka-u.ac.jp

あらまし　大規模・複雑化が進むネットワークシステムに対して環境の変化に対する適応性を備えさせるという目的から、我々はこれまでに管理型自己組織化制御に着目して研究を行ってきた。管理型自己組織化制御は、大規模なネットワークにおいて自己組織化制御で特に問題となる、ネットワークの機能創発を制御できない点、大域的な最適性が保証できない点、広範囲な環境変動後のシステム状態の収束に長時間を要する点の解決を図るものである。個々の端末は局所情報に基づいて動作し、外部のコントローラーがシステムを観測し一部の端末に対して制御を加える事で、自己組織化制御の利点を残したまま、上記の問題点を解決する。本稿で対象とする広範囲な環境変動に対する適応速度の遅さは、自己組織化制御における局所情報にもとづく局所的な行動決定という特徴がまねくものであるが、一方でこの特徴は局所的な環境変動に対する適応性をシステムにもたらすものでもある。我々は制御理論の知見を用いることで、システムの状態の収束速度を高速化するように、コントローラーが自己組織化制御に対する制御入力を決定する方法を提案する。線形の自己組織型ネットワークシステムを対象に、大規模な環境変動に対する適応性が向上することを示した。

キーワード　自己組織化制御、線形ネットワークシステム、管理型自己組織化制御、ロバスト制御

[Invited Talk] Enhancing Adaptability of Self-organizing Network Systems

Daichi KOMINAMI†, Kenji KASHIMA††, Tomoaki HASHIMOTO†††, and Masayuki MURATA††††

† Graduate School of Economics, Osaka University
†††† Graduate School of Information Science and Technology, Osaka University
††† Graduate School of Engineering Science, Osaka University
†† Graduate School of Informatics, Kyoto University
E-mail: †d-kominami@econ.osaka-u.ac.jp, ††kashima@amp.i.kyoto-u.ac.jp, †††thashi@sys.es.osaka-u.ac.jp,
††††murata@ist.osaka-u.ac.jp

Abstract　We have focused on controlled self-organization to provide more adaptability for large-scale network systems. Self-organization protocols, which are based on local determination with local information, shows high scalability and robustness, however, for practical applications in much larger networks, they have some disadvantages such as difficulty of controlling functional emergence and slow convergence speed for wide range of environmental changes. In this paper, we enhance convergence speed of self-organization protocols using control theory knowledge. We show that this realizes the faster convergence of a linear self-organized network system.

Key words　Self-organization, linear network system, controlled self-organization, robust control.

1. まえがき

今後ますます大規模化、複雑化が進む通信ネットワークにおいて、その適切な制御および管理は、非常に重要ながら困難な課題である。この課題に対して、我々はこれまでに管理型自己組織化制御 [1] に基づくネットワークシステムを提案し、その有効性を示してきた [6]。本稿ではその中で未解決であった、広範な環境変化に対する適応速度の遅さ、すなわち低い適応性の解決を図る。

ネットワークを構成する端末の種類や数が膨大な数となり、その利用形態や利用環境も様々に変化すると考えられる中で、高い拡張性、適応性、頑強性を有する制御技術として、自己組織化制御に関する研究が注目されてきた [2]。自己組織化の原理を応用した自己組織化制御では、システムの個々の要素がシステム全体についての情報を持つことなく、局所情報による行動決定を行い、全体としての機能が創発される。制御の局所性により、局所的に発生した環境変動の影響がシステム全体に波及しにくく、かつ、局所的な行動決定の変更もシステム全体に影響を波及しにくい [3]。そのため、局所範囲での環境変動に対しては局所的な対応を行うことで、即座な適応が可能である。一方で、広範囲での環境変動が生じた場合は、システムが到達すべき状態への遷移を個々の要素の行動決定にゆだねているために、システムが安定状態に到達するまでに制御を何度も繰り返す必要がある。多くの場合は、個々の要素が収集する情報量・行動決定のための計算量を小さく抑えることで、短い制御周期を設定できるため、これによって安定状態までの到達時間を短くしているが、ネットワークシステムが大規模化するほどに要する時間は長くなる。これまでに自己組織化制御の適応性については様々に研究がなされているが、そのほとんどは局所的な変動に対する解決を対象としたものである [4]。

完全に局所情報のみから行動決定を行う自己組織化制御では、そのボトムアップによる設計から、システム全体の管理の困難さ、最適な動作保証の困難さ、大域的な環境変化への適応速度の遅さ、などの問題が生じることが既に指摘されており [5]、管理型自己組織化制御はこれらの問題を解決する可能性をもつものとして、文献 [1] で提案された概念である。管理型自己組織化制御では、自己組織化制御に基づくシステムに対して、その外部にシステムの監視と制御を行うコントローラーを追加し、コントローラーによる適切な制御入力をシステムに与えることで、従来の自己組織化制御のみでは得られなかった特徴をシステムに与えることができる。

我々は文献 [6] において、センサーネットワークにおける経路制御を対象に、管理型自己組織化による機能創発のコントロールと大域的な最適化を実現したが、広範囲の環境変動に対する適応性の低さは依然として残る課題であった。広範囲の環境変動への適応が遅いということは、自己組織型のシステムに対してコントローラーが制御を行う場合、システム全体の状態が収束するまでに大きな時間を要することを示しており、そもそも管理型自己組織化制御を用いる上で大きな問題となっている。

そこで収束速度の高速化を実現するために、制御理論の知見を用いることで、コントローラーがネットワークに与える制御入力として適切な値を決定し、その入力を用いて制御を行う手法を提案する。提案手法の実現可能性を示すために、まず線形の自己組織型ネットワークシステムを対象として、広範囲の環境変動に対する適応性の向上が可能であることを示すのが本稿の目的である。文献 [6] において対象としていたポテンシャルルーティングは、物理学的なポテンシャル場をネットワーク上に擬似的に再現し、場の勾配を用いて経路を決定する経路制御手法である。このポテンシャルの場の構築は、個々の端末が局所情報のみに基づいた線形の更新則に従ってポテンシャルを計算することでなされる。このようなシステムに対して、ロバスト制御 [7]（あるいは最適制御）と呼ばれる制御理論の適用が可能である。ロバスト制御とは、制御対象システムの数式モデルが多少不確かな場合にも目的の状態に近づけるための最適な入力を決定する制御理論である。特に提案手法の利点として、システムのモデル推定を行うことで、ネットワークシステムに関する情報収集量を抑えつつ、高速な収束が可能な点がある。

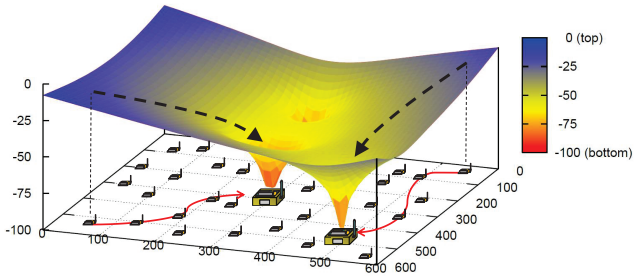


図 1　ポテンシャルルーティング

本稿ではまず 2. でポテンシャルルーティングの概要を説明する。3. において、自己組織型ネットワークシステムを制御対象としたときの、ロバスト制御に基づく制御器の設計方法について述べる。4. で、広範囲に環境変動が発生した際の、自己組織型ネットワークシステムの適応性向上を、計算機シミュレーションを用いて示す。最後に 5. で本稿のまとめを述べる。

2. ポテンシャルルーティングの概要

ポテンシャルルーティングの概要を図 1 に示す。図は無線センサーネットワークにおけるポテンシャルルーティングの様子を表しており、センサーノードと呼ばれる無線端末が、観測したデータを宛先となるシンクノードにマルチホップで届けている（実線矢印がデータの流れを表す）。各センサーノードおよびシンクノードはポテンシャルと呼ばれるスカラー値を所持しており、図では便宜上、ノードの存在しない箇所については補完を行った擬似的なスカラーポテンシャル場を示している。このとき、無線通信が可能な範囲内に存在する他のノードとのポテンシャル差をグラディエントと考え、グラディエントの大きさに従って中継先を選択するのがポテンシャルルーティングと呼ばれる経路制御手法である。本稿では図 1 に示すように、グラディエントが負となる方向にデータが流れることとしており（破線矢印）、シンクノードが極小の値を持つようにポテンシャ

ル場を構築することで、ノードはデータをいずれかのシンクノードに到達させることができる。

2.1 ノードごとのポテンシャル更新則

ノードの更新則は式 (1) を用いることとする [8]。式 (1) は、次時点のポテンシャルの計算のために現時点および一時点前の情報を利用しており、現時点の情報のみを利用する手法（文献 [6] など）と比較して収束速度を高速化している。二時点前、三時点前と過去の情報を増やすことで、収束速度の向上を図ることができるが、あくまでも局所的な情報に基づく考え方であり、本稿の検討内容とは異なるためここでは検討外とする。

$$u_i^{t+1} = (\alpha+1)u_i^t - \alpha u_i^{t-1} + \beta\Big(\sum_{j\in nb(i)}\{u_j^t - u_i^t\} - f_i^t\Big) \quad (1)$$

式 (1) において、u_i^t はノード i の時刻 t でのポテンシャルを表す。α, β はパラメーターであり、α はポテンシャルの更新時に現時点の情報に加えて前時点の情報をどの程度含めるか、β は現時点で結合している（通信範囲内にある）ノードからどの程度の影響を受けるかを表す。また、各ノードのフロー発生レートを f_i^t で表し、センシングした情報をデータパケットとして発生するセンサーノードの場合は正の値となり、データを収集するシンクノードには負の値を設定しておく必要がある。

以上の更新則により構築されるポテンシャル場は、初期状態への依存はあるものの基本的に一意に定まる。また、ポテンシャルの値自体は特に意味を持たないものの、グラディエントの大きさは、フロー発生レートを満たすような次ホップへのフロー割り当て量に比例する値となっている。そのため、グラディエントの大きさに比例して次ホップへ流すフロー量を決定することで、シンクノードに届くフローがあらかじめ定めたレートに従うことになる。

3. ネットワークダイナミクスと制御系設計

本稿で想定するネットワークシステムについて説明する。ここではマルチシンク型のセンサーネットワークを対象とし、システム外部に存在するコントローラーはシンクノードに対してのみ制御入力を与える。コントローラーはシンクノードを介して、制御に必要なネットワークの情報を収集する（図 2）。後の式で用いるために収集する必要がある情報は、各時点におけるネットワークの隣接行列、各ノードのポテンシャル、各ノードのフロー発生レートである。ネットワークの隣接行列に変更があった際には制御系の再設計が必要となるが、本稿ではトポロジーの変化の影響については生じないものと仮定する。コントローラーが観測を行わないノードについては、観測した情報に基づいてそのノードの状態を推定して制御入力を決定する。以降では、推定に用いるネットワークダイナミクスとコントローラーの制御量の決定方法について説明する。

3.1 ネットワークダイナミクス

式 (1) は、個々のノードのポテンシャル更新則を記述したものであり、三時点分の情報が含まれている。一階の微分形式で記述される状態方程式の形式でシステムを表現するため、二時点の情報のみを用いてネットワーク全体のダイナミクスを表現

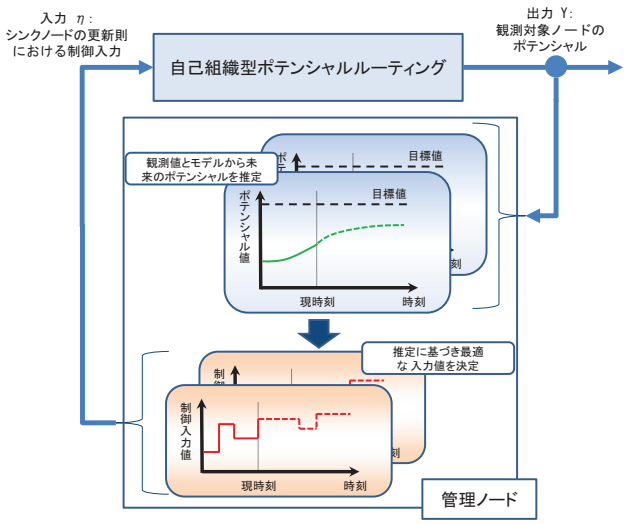


図 2　システムモデル

する。そこで、ベクトル $\boldsymbol{u_i^t}$ を以下のように定義する。

$$\boldsymbol{u_i^t} = \begin{bmatrix} u_i^t \\ u_i^{t-1} \end{bmatrix} \quad (2)$$

すでに述べたように、u_i^t はノード i の時刻 t でのポテンシャルである。このように定義した $\boldsymbol{u_i^t}$ を用いることで、式 (1) は以下の式 (3) で表すことができる。

$$\boldsymbol{u_i^{t+1}} = A\boldsymbol{u_i^t} + A_0\sum_{j\in nb(i)}(\boldsymbol{u_j^t} - \boldsymbol{u_i^t}) + Bf_i \quad (3)$$

このとき A, A_0, B はそれぞれ以下で表される。

$$A = \begin{bmatrix} 1+\alpha & -\alpha \\ 1 & 0 \end{bmatrix},\ A_0 = \begin{bmatrix} \beta & 0 \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} -\beta \\ 0 \end{bmatrix}$$

式 (3) は、単に展開すれば、式 (1) と等価であることが分かる。ここでネットワーク中のノード数を N とし、$\boldsymbol{u_1^t}$〜$\boldsymbol{u_N^t}$ を並べた行列 $\boldsymbol{U^t} = [\boldsymbol{u_1^t}\ \boldsymbol{u_2^t}\ \cdots\ \boldsymbol{u_N^t}]^T$ を用いると、式 (3) に基づくネットワークダイナミクスは式 (4) で表すことができる。

$$\boldsymbol{U^{t+1}} = \big(I_{N\times N}\otimes A - \Gamma\otimes A_0\big)\boldsymbol{U^t} + \big(I_{N\times N}\otimes B\big)F^t \quad (4)$$

このとき、$I_{N\times N}$ はサイズ N の単位行列、Γ はグラフラプラシアン、F^t は $[f_1^t\ f_2^t\ \cdots f_N^t]^T$ をそれぞれ表している。また、$\otimes$ はクロネッカー積を表す。本稿では、各ノード i の時刻 t におけるフローレート f_i^t は、断らない限りは一定かつ既知と仮定する。

3.2 ロバスト制御系設計

式 (4) に対して、ポテンシャルへの制御入力を表す項 $(E\otimes[1\ 0]^T)\eta^t$ を追加したのが式 (5) である。

$$\boldsymbol{U^{t+1}} = \big(I_{N\times N}\otimes A - \Gamma\otimes A_0\big)\boldsymbol{U^t} + \big(I_{N\times N}\otimes B\big)F^t + (E\otimes C)\eta^t \quad (5)$$

ここで、E は制御入力を与えるノード（本稿ではシンクノードであり、個数を m 個とする）を表す $N\times m$ の行列であり、E

の各列において制御の対象となるノードに対応する行は 1、それ以外は 0 となっている。η^t は実際の制御入力値を表す $m \times 1$ の行列である。また、$C = [1\ 0]^T$ としている。以降では、このような η^t として、最適な値を求める方法について説明する。

コントローラーはネットワークの情報を収集しており、Γ, F^t が既知であれば、現時点における全てのノードのポテンシャルの収束値を求めることが可能である。そのようなポテンシャルの収束値を、$\bar{\boldsymbol{U}}$ と表記する。ここで、$\bar{\boldsymbol{U}}$ は、式 (4) において、$\boldsymbol{U^{t+1}} \leftarrow \bar{\boldsymbol{U}}$、$\boldsymbol{U^t} \leftarrow \bar{\boldsymbol{U}}$ と代入することで、式 (6) のように求められる。

$$\bar{\boldsymbol{U}} = \big(\Gamma \otimes A_0\big)^{-1}\big(I_{N\times N} \otimes B\big)F^t \tag{6}$$

このとき、行列 $\tilde{\boldsymbol{U}}^{\boldsymbol{t}} = \boldsymbol{U^t} - \bar{\boldsymbol{U}}$ とおくと、$\tilde{\boldsymbol{U}}^{\boldsymbol{t}}$ の更新則は、式 (5) へ $\boldsymbol{U^t} = \tilde{\boldsymbol{U}}^{\boldsymbol{t}} + \bar{\boldsymbol{U}}$ を代入することによって得られ、式 (7) が得られる。

$$\tilde{\boldsymbol{U}}^{\boldsymbol{t+1}} = \big(I_{N\times N} \otimes A - \Gamma \otimes A_0\big)\tilde{\boldsymbol{U}}^{\boldsymbol{t}} + \big(E \otimes C\big)\eta^t \tag{7}$$

コントローラーが $\tilde{\boldsymbol{U}}^{\boldsymbol{t}}$ を観測できる場合、式 (8) で定められるコスト関数 J を最小化することで最適な η^t が得られる。

$$J = \sum_{t=0}^{\infty} \left(||\tilde{\boldsymbol{U}}^{\boldsymbol{t}}||^2 + r||\eta^t||^2\right) \tag{8}$$

ここで述べる最適な η^t とは、$\tilde{\boldsymbol{U}}^{\boldsymbol{t}}$ の絶対値の無限時間先に渡っての累積和を最小化するような値であるが、そのような η^t を入力として与えると、急激なポテンシャルの変化が生じる可能性がある。そこで、制御パラメーター r を適切な値に設定することで制御入力の大きさを抑えることとする。r を 0 に近づけるほどポテンシャル場の収束を重視し、大きく設定するほど制御によるポテンシャル場の変化抑制を重視する。式 (7) のようなダイナミクスで表現されるシステムでは、$\eta^t = K\tilde{\boldsymbol{U}}^{\boldsymbol{t}}$ という形式で η^t が与えられることが知られている。最適なフィードバックゲインを与える行列である K については、式 (4) で表されるシステムが安定であれば求められることが知られている。本稿においては、例えば Γ の結合が強い場合に不安定化が生じる可能性があり、これを防ぐためには β を十分小さく設定する必要がある。

コントローラーが $\tilde{\boldsymbol{U}}^{\boldsymbol{t}}$ を観測せず、一部のノードのみを観測してそれ以外のノードのポテンシャルを推定し、制御器の設計を行う場合には、以下の $\boldsymbol{Y^t}$ によって表される観測対象となるノードのポテンシャルを用いて $\tilde{\boldsymbol{U}}^{\boldsymbol{t}}$ の推定を行う。

$$\boldsymbol{Y^t} = (H^t \otimes I_{2\times 2})\tilde{\boldsymbol{U}}^{\boldsymbol{t}} \tag{9}$$

H^t は E^t と同様で、こちらは観測の対象となるノードを 1、それ以外を 0 で表す。$\boldsymbol{Y^t}$ の観測によって推定されるネットワーク全体のポテンシャルを $\tilde{\boldsymbol{U}}^{\boldsymbol{t}}_{\boldsymbol{est}}$ とするとき、以下の式 (10) および式 (11) で式 (8) の J を最小化するような A_c, B_c, C_c を導出することが目的となる。

$$\tilde{\boldsymbol{U}}^{\boldsymbol{t+1}}_{\boldsymbol{est}} = \boldsymbol{A}_c\tilde{\boldsymbol{U}}^{\boldsymbol{t}}_{\boldsymbol{est}} + \boldsymbol{B}_c\boldsymbol{Y}^t \tag{10}$$

$$\eta^t = \boldsymbol{C}_c\tilde{\boldsymbol{U}}^{\boldsymbol{t}}_{\boldsymbol{est}} \tag{11}$$

表 1　シミュレーションにおけるパラメーター

Parameter	Value
T_i	50 [s]
T_c	50 [s]
α	0.4
β	0.02
r	0.0001,10

このような A_c, B_c, C_c は、半正定値計画問題を解くことで得られることが知られている。以上の制御入力の決定方法はすべてのノードが同期して動作することを前提としており、実際のネットワークのような非同期で動作する場合にも正しく動作することをシミュレーションにより次章で示す。

4. シミュレーション評価

コントローラーによる制御によるポテンシャル場の収束速度の変化を評価するために計算機シミュレーションを行った。本稿では特に、ポテンシャル場の収束にのみ着目するが、ポテンシャル場の収束が向上することで実際のフローも安定化することが期待される。シミュレーションには、Visual C++で作成したイベントドリブンモデルのパケットレベルシミュレーターを用いた。

4.1 シミュレーション設定

ネットワークモデルとして図 3 に示すトポロジーを利用し、4 台のシンクノードと 50 個のセンサーノードがポテンシャル場を構築する。個々のセンサーノードは周囲のノードとポテンシャルの情報交換を、コントローラーはシンクノードから 2 ホップ内に限定してポテンシャル情報の収集を行う。

ポテンシャルの収束速度の向上の効果を調べるために、シミュレーションにおける時刻 10,000 s に、フローの流入出行列である F^t を変更することで、ポテンシャル場の再構築を行い、その際のポテンシャルの変化を調べる。具体的には、シミュレーション開始から時刻 10,000 s までは、ノードのフローレートを

$$F^t = [0.5, \cdots, 0.5, 1.5, \cdots, 1.5, -12.5, -12.5, -12.5, -12.5]^T$$

とし、時刻 10,000 s 以降は

$$F^t = [1.5, \cdots, 1.5, 0.5, \cdots, 0.5, -12.5, -12.5, -12.5, -12.5]^T$$

に変更する。ここで、行列 F^t の最後から 4 つの要素が、それぞれシンクノードを表している。

シミュレーションにおけるパラメーターは表 1 に示すとおりである。コントローラーによるポテンシャル制御周期である T_c およびポテンシャルの更新周期 T_i はいずれも 50 s としている。各ノードのポテンシャルの更新は非同期に行われ、50 s 毎に一度、周囲のノードから現在のポテンシャルを取得し、その値を元に自身のポテンシャルを計算して更新する。また、コントローラーによる制御と、上記のポテンシャル更新についても非同期に行われる。$\tilde{\boldsymbol{U}}^{\boldsymbol{t+1}}_{\boldsymbol{est}}$ の計算のための A_c, B_c, C_c については、MATLAB の $dhinflmi$ 関数を用いることで設計を行った。

4.2 シミュレーション結果

図 3 に示したネットワークトポロジーにおいて、ロスおよび

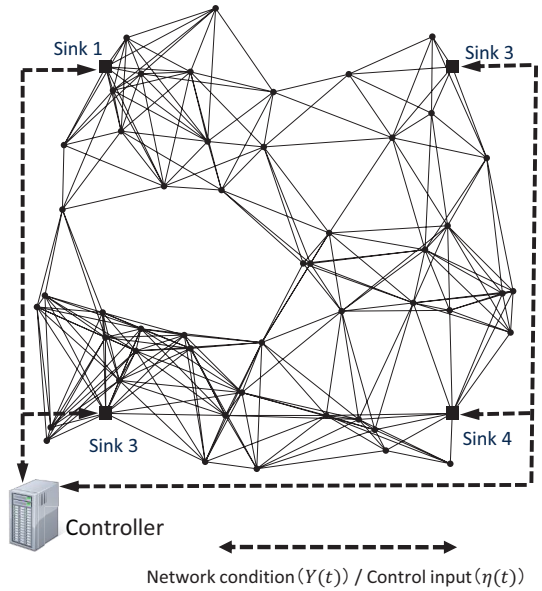


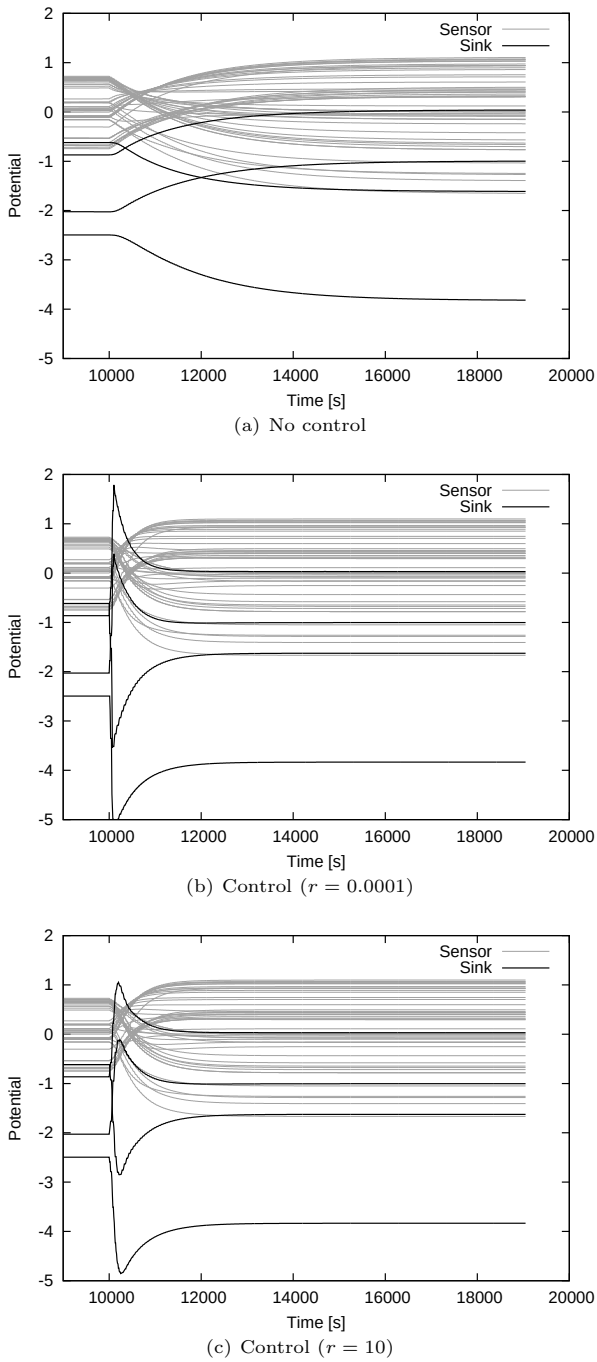
図 3　ネットワークモデル；点がセンサーノードを、四角がシンクノードを、実線がノード間のリンクを、破線がシンクノードとコントローラー間のリンクをそれぞれ表している。コントローラーによる制御対象は、全シンクノード（4 台）である

遅延なく、コントローラーが情報を観測できることを仮定した場合に、ノードのポテンシャルの変化を調べた結果が図 4 である。

図 4 は、フローレートが切り替わる時刻 10,000 s 以降の、各時刻におけるノードのポテンシャルを示しており、濃色線がシンクノードのポテンシャル、淡色線がセンサーノードのポテンシャルである。制御入力値に対する制約の大きな図 4(c) の方が、図 4(b) でのシンクノードのポテンシャルよりも 10,000 s 直後の変化は緩やかであるものの、制御を行わない図 4(a) と比較して、制御を行う図 4(b)、 4(c) の収束速度が向上していることが分かる。

図 4(b)、 4(c) は、観測する情報を 2 ホップに限定したものであり、推定モデルを用いることで制御入力を決定している。さらに、T_c および T_i を同じ値として計算した本結果では、ノード同士、またノードとコントローラーが非同期に動作しており、提案手法において望んだ結果が得られることが示された。以上より、ロバスト制御を線形の自己組織型ネットワークシステムに用いることによって適応性の向上が実現できることが示された。

図 5 はコントローラーが情報を観測する際に、ロスが発生する場合の、各ノードのポテンシャルの変化を調べた結果である。ここではコントローラーはシンクノードを介して情報を観測するため、通常のセンサーノードからシンクノードへのトラフィックと衝突する場合があり、衝突の際にパケットロスが発生することとしている。得られた結果から、コントローラーによる制御が、ポテンシャルの収束速度の向上をもたらしていることが分かる。特に、ロスが発生する場合にはコントローラー設計時には未知の誤差が生じることになるが、この際にも収束速度の向上が実現できることが示された。



(a) No control

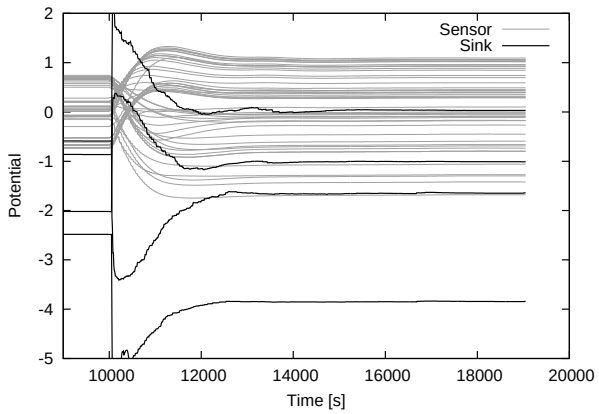
(b) Control ($r = 0.0001$)

(c) Control ($r = 10$)

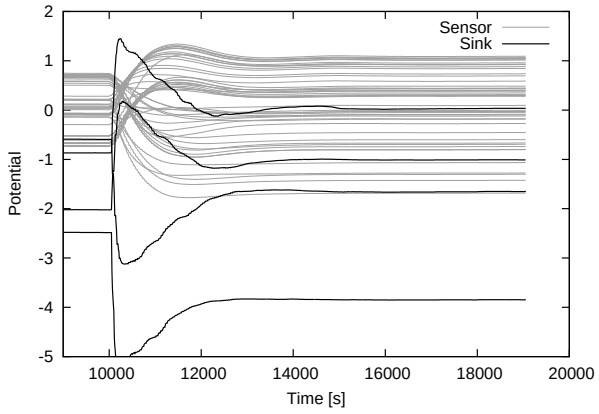
図 4　制御の有無とポテンシャルの収束速度向上

5. ま と め

本稿では、大規模複雑化が進むネットワークのために、制御理論の知見を用いた管理型自己組織化制御を提案し、大規模な環境変動に対する適応性向上が実現可能であることを示した。センサーネットワークを対象としたポテンシャル場に基づく経路制御手法は、線形の自己組織型ネットワークシステムであり、これを一例として、個々の端末はあくまでも局所情報に基づいて動作しながら、コントローラーが一部の端末に最適な制御入力を与えることで、システム全体の収束時間を高速化できることを示した。しかしながら、今回の検討には様々な前提をおいており、特に線形のシステムであること、陽にシステム



(a) $r = 0.0001$



(b) $r = 10$

図 5　コントローラーの情報観測にロスが含まれる場合

の目標値が定まるといった仮定は、一般的な自己組織型のネットワークシステムへの適用に向けての大きな制限となる。また、センサーネットワークのようなトポロジーの変化が比較的発生しやすい環境では、トポロジーが変わるごとの制御系の再設計が必要となる点についても問題となる。また、本稿で用いた MATLAB の $dhinflmi$ 関数は、端末数を増加させた時に実時間での計算が困難となるケースがあり、この点についても解決が必要となる。現在はシステムモデルの状態数削減、コントローラーの分散配置に取り組んでおり、非線形システムへの適用可能性を示すこと、トポロジー情報の適切な管理、利用方法の検討等が今後の課題である。

文　献

[1] J. Branke, M. Mnif, C. Müller-Schloer, H. Prothmann, U. Richter, F. Rochner, and H. Schmeck, “Organic Computing–Addressing Complexity by Controlled Self-Organization,” Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA), pp.185–191, 2006.

[2] F. Dressler, Self-Organization in Sensor and Actor Networks, Wiley, 2007.

[3] M. Sugano, Y. Kiri, and M. Murata, “Differences in Robustness of Self-Organized Control and Centralized Control in Sensor Networks Caused by Differences in Control Dependence,” Proceedings of the International Conference on Systems and Networks Communications (ICSNC), pp.145–150, Oct. 2008.

[4] H. Zhang, J. Llorca, C.C. Davis, and S.D. Milner, “Nature-Inspired Self-Organization, Control, and Optimization in Heterogeneous Wireless Networks,” IEEE Transactions on Mobile Computing, vol.11, no.7, pp.1207–1222, July 2012.

[5] C. Müller-Schloer, H. Schmeck, and T. Ungerer, Organic Computing–A Paradigm Shift for Complex Systems, Birkhäuser, 2011.

[6] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Controlled and Self-Organized Routing for Large-Scale Wireless Sensor Networks,” submitted to ACM Transactions on Sensor Networks, vol.10, no.1, pp.1–12, Nov. 2013.

[7] K. Zhou, J.C. Doyle, K. Glover, et al., Robust and Optimal Control, Prentice Hall New Jersey, 1996.

[8] A. Sheikhattar and M. Kalantari, “Fast Convergence Scheme for Potential-Based Routing in Wireless Sensor Networks,” Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), pp.1980–1985, April 2013.