計測自動制御学会東北支部 第 303 回研究集会 (2016.7.15)
資料番号 303-9

IP ネットワークを介した DC モータ制御システムの通信遅延時間分布の変化に適応した一制御法の実装

Implementation of a Control Method for DC Motor Control System through IP Network with Changing Network Conditions

○市川拓*，松尾健史*，三浦武*，田島克文*
○Taku Ichikawa*, Kenshi Matsuo*, Takeshi Miura*, Katsubumi Tajima*
*秋田大学
*Akita University

キーワード: 遠隔制御(remote control)，DC モータ(DC motor)，IP ネットワーク(IP network)，遅延時間(delay time)，変化点検出(change point detection)

連絡先：〒010-8502　秋田県秋田市手形学園町 1-1　秋田大学大学院　理工学研究科
松尾　健史，Tel.：(018)889-2332，Fax.：(018)837-0406，E-mail：matsuo@ipc.akita-u.ac.jp

1. はじめに

現在，情報通信端末のみならず，様々な物がインターネットすなわち IP ネットワークを用いてネットワーク化されている．これに関連し，IP ネットワークを介した遠隔制御が注目されており，これに関する研究も行われている [1].

制御対象とするアクチュエータの基本的なものとして，サーボモータがあげられる．このサーボモータは位置，速度等を制御する用途に使用される．その代表の一つとして DC モータがあげられ，本研究ではこの DC モータを用いて以後実験を行う．

IP ネットワークを介して制御システムを構築することの利点として，既存の IP ネットワークにコントローラ，アクチュエータ等の機器を取り付けるのみで容易かつ安価に制御システムの構築が可能である [1]といった点があげられる．

しかし，IP ネットワークは制御を前提に構築されていないために，これを考慮せずに制御を行うと，制御性能の劣化が発生する問題がある [2].

この劣化は通信遅延時間やその揺らぎにより引き起こされる．そこでこの問題の改善策としてゲインスケジューリング [2]，ジッタバッファ，および冗長伝送を用いる手法 [3]等が提案されている．

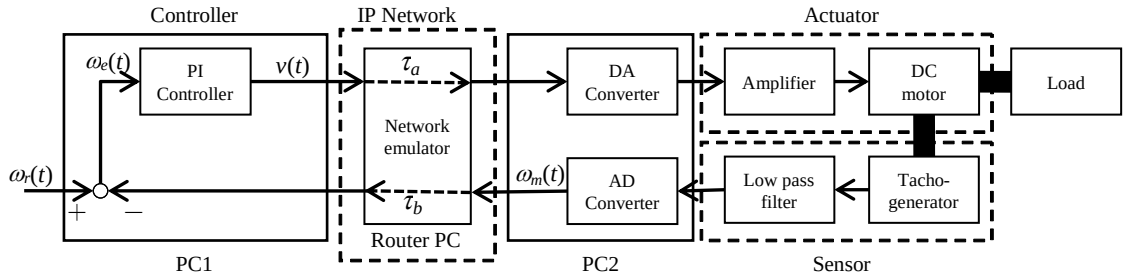


図 1　IP ネットワークを介した DC モータ遠隔速度制御システムの構成図

Fig. 1　Configuration of speed control system for DC motor through IP network

しかし，これらの手法は通信遅延時間の分布が大きく変化する場合において十分に対応しているとは言えない．そのため文献 4)では，ChangeFinder[5)]と呼ばれる変化点検出アルゴリズムを用いて， 遅延分布の変化を検出し，ゲインスケジューリングを行う手法を提案し，またその有効性が数値シミュレーションで示された．

しかし，文献 4)では数値シミュレーションのみでしか評価していない．そこで本研究では，この制御法を実際の制御システムへ実装し，IP ネットワークを介して対象とする DC モータの速度制御実験を行うことで，このシステムの有用性について検証する．

2. 制御システム

本研究では IP ネットワークを介して DC モータを制御するシステムを構成し，この構成図を図 1 に示す．

図 1 において$\omega_r(t)$は目標回転速度[min^{-1}], $\omega_m(t)$は実際の DC モータの回転速度[min^{-1}], $\omega_e(t)$は目標回転速度と実際の DC モータの回転速度との偏差[min^{-1}]，$v(t)$は DC モータへの印加電圧[V]である．τ_a，τ_b はそれぞれ IP ネットワークを介した際の片道通信遅延時間[ms]である．また $L = \tau_a + \tau_b$[ms]が往復遅延時間であり，RTT はこの値となる．

構成図において，PC1 は PI 制御を行うコントローラの役割を果たし，PC2 は PC1 から受信した印加電圧を DA 変換によってモータに印加し駆動させ，センサで検出した回転速度を AD 変換によって取得する．すなわちドライバの役割を果たす．

IP ネットワーク部には Router PC があり，PC1 と PC2 の通信時にデータを中継する．また PC 間の通信プロトコルには，リアルタイム性に優れる UDP(User Datagram Protocol)を用いる．

Router PC にはネットワークエミュレータ netem[6)]がインストールされている．実際の IP ネットワークではネットワークの負荷や通信環境などにより，通信遅延時間やその揺らぎが変動するため，それらを任意の条件で発生させることは困難である．そこで本研究では netem で通信遅延時間分布の平均[ms]と標準偏差[ms]を設定し，模擬的にネットワーク環境を構築して実験を行う．

使用した PC の仕様を表 1 に示す．また LAN カードは 1Gbps 1000BASE-T のものを使用し，コンパイラは各 PC ともに gcc version 4.8.2 であり，C++で開発した．また

表 1　機器の仕様

Table 2　Specification of machine

PC1	
OS	Linux kernel 3.14.7 Fedora 20
CPU	Intel core i5-3330 3.00 GHz
PC2	
OS	Linux kernel 3.13.9 Fedora 20
CPU	Intel core i5-3330 3.00 GHz
PC3	
OS	Linux kernel 3.13.9 Fedora 20
CPU	Intel core i5-3330 3.00 GHz

表 2　DC サーボモータの仕様

Table 1　Specification of DC servo motor.

Rated output	11 W
Rated voltage	24 V
Rated current	1.25 A
Rated speed	3000 min^{-1}

制御対象として，山洋電機社製 DC サーボモータ R301T-011 を使用する．その仕様を表 2 に示す.

またタコジェネレータは 3 V/1000 min^{-1} のものを使用し，慣性負荷には 2.5×10^{-5} N・m・s^2/rad のものを装着する．またサンプリング時間は 1 ms とする.

3. 遅延時間の変化の検出に関する実験

通信遅延時間の変化点の検出を行うため，文献 4)においても用いられたネットワーク分野で使用されている変化点検出を用いる．これにより実際の制御システムにおいてもリアルタイムに検出が可能であるかの検討を行う.

本章では通信遅延時間が正規分布の場合について実験を行う．なお制御時間を 4000 ms とし，2000 ms 時に通信遅延時間が切り替わるものとする.

3.1 通信遅延時間の分布の変化検出

本研究では、通信遅延時間分布の変化を検出するために、ウイルスなどによるトラフィックの時系列的な増大等を即時検知可能な ChangeFinder[5)]を用いた.

これは, 自己回帰モデル(AR モデル : Auto Regression Model)を用いて動作し, サンプリング時間ごとに通信遅延時間を取得した後, 2 段階の学習により分布の変化度合であるスコア値を計算する．このとき，第 1 段階の学習で外れ値を算出し，第 2 段階の学習で本質的に変動を検出する．この ChangeFinder により出力されるスコア値が大きいほど, 変化度合が大きいこととなる.

つまり, スコア値が大きく変化したとき, 通信遅延時間の分布が変化したことがわかる．このフローチャートを図 3 に示す．入力 L_i は各サンプリング時間 i における RTT の値である．また p_{i-1} は L_1，…，L_{i-1} から学習された確率密度関数であり, q_{i-1} は y_1, …, y_{i-1} から学習された確率密度関数である．ChangeFinder による時刻 t における最終的なスコア値は

$$\mathrm{Score}(t)=\frac{1}{T'}\sum_{j=t-T+1}^{t}(-\ln q_{j-1}(y_j)) \quad (1)$$

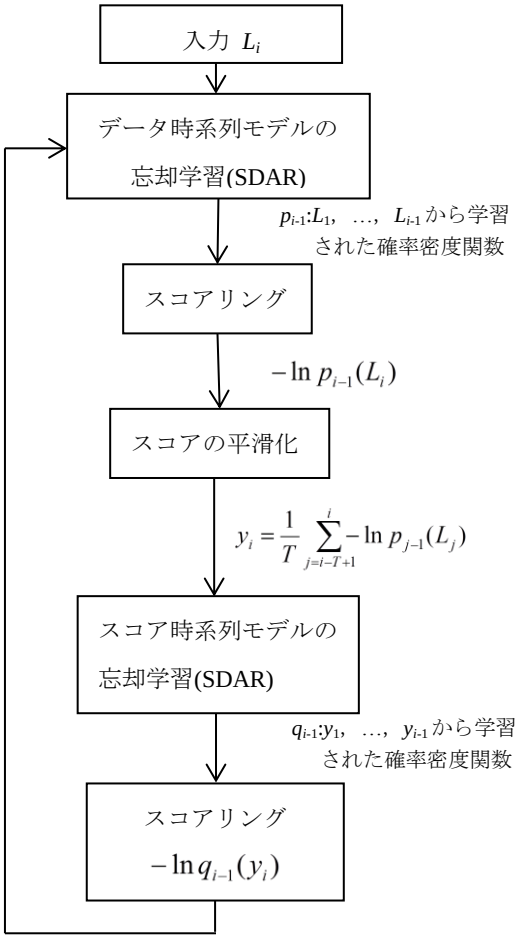


図 3　ChangeFinder のフローチャート [5)]

Fig. 3　Flowchart of ChangeFinder[5)]

となる．前述したように，この値が大きくなったとき通信遅延時間の分布が変化したことがわかる．詳しくは文献 5)を参照されたい．

なお，ChangeFinder の各種パラメータは，忘却パラメータをγ =0.5、AR モデルの次数を k =1、第 1 段階学習によって得られた移動平均平滑化の区間長を T=7，さらに第 2 段階学習によって得られた移動平均平滑化の区間長を T'=4 のように設定した．ここでは文献 4)と同様の値を用いた．

この ChangeFinder に RTT を入力すると遅延分布の変化を検出できる．そこで，制御法の詳細は次の章で述べるが，スコア値がある一定値以上を出力したとき，遅延分布の変化があったとみなし，その変化した遅延分布に対する適切なゲインに調整する制御法を本実験では用いる．つまり，スコア値にしきい値を設け，しきい値以上のスコア値になった場合に，ゲインスケジューリングの手法を用いる．

しかしこの制御法の実装を行う前に，実際のシステムにおいて，RTT から遅延分布の変化を検出できるかの実験を行う．

3.2　通信遅延時間の変化点検出実験

通信遅延時間の変化の大きさをリアルタイムで実際のシステムにおいて検出できるかを調べる．前半および後半の分布が共に正規分布である場合において表 3 の条件で実験を行う. Experiment 1 は遅延時間の平均が大きくなる場合，2 は標準偏差が大きくなる場合，3 は平均が小さくなる場合に，そして，4 は標準偏差が小さくなる場合でそれぞれ実験を行う．

また図 4 は Experiment 1 の 3 つ目の実験で，前半の遅延分布の平均が 20 ms，標準偏差が 4 ms で，後半の分布の平均が 120 ms，標準偏差 4 ms のものを示している．このように，設定する遅延分布を変化させ実験を行う．

そして今回行う実験では図 1 におけるτ_bの遅延時間のみ変更し，τ_a=0 とした．また実験システムの制約により，遅延時間の分布の変更は，切り替える時刻付近で手動にて行うため，切り替え点が多少前後することについては注意されたい．

表 3　遅延時間分布の実験条件

Table 3　Experiment condition of delay distribution

	The distribution of the first half period		The distribution of the latter half period	
	mean value [ms]	standard deviation [ms]	mean value [ms]	standard deviation [ms]
Experiment 1	20	4	40	4
			80	
			120	
Experiment 2	80	4	80	12
				18
				24
Experiment 3	100	4	20	4
			40	
			80	
Experiment 4	100	28	100	4
				12

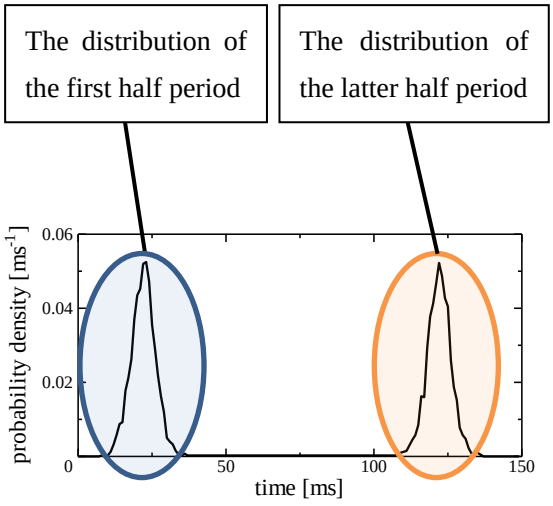
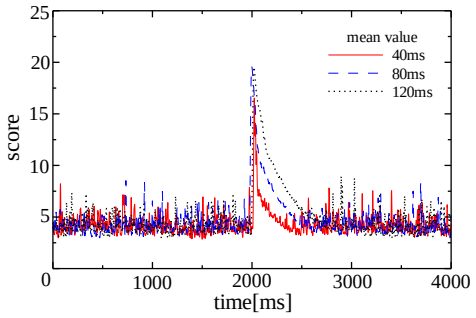
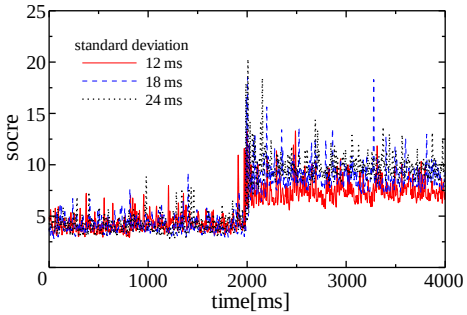


図 4　RTT の確率密度分布

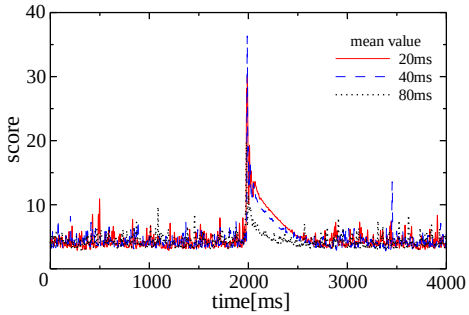
Fig. 4　Probability density distribution of RTT



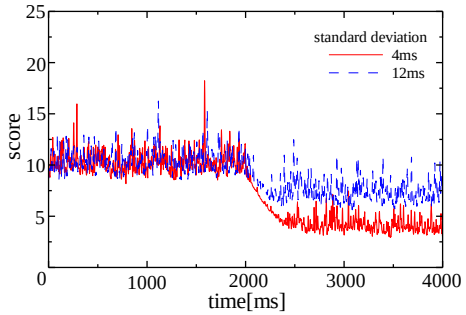
(a) Experiment 1



(b) Experiment 2



(c) Experiment 3



(d) Experiment 4

図 5　遅延時間の変化点検出

Fig. 5　Change point detection of delay time

表 4　制御ゲイン値の実験条件

Table 4　Experiment condition of control gain value

Delay time[ms]	Gain
$L < 120$	K_P=0.0014，K_I=0.0068
$120 \le L < 140$	K_P=0.0014，K_I=0.0062
$140 \le L < 160$	K_P=0.0014，K_I=0.0052
$160 \le L < 180$	K_P=0.0013，K_I=0.0045
$180 \le L$	K_P=0.0013，K_I=0.004

図 5 でこの結果を示す. (a), (b), (c),および (d)がそれぞれ，Experiment 1, 2, 3,および 4 の結果を示している.

Experiment 1 および 3 の結果より, 平均値の変化に対しては変化点が検出されることが分かった．また，遅延分布を変更した後のスコア値の大小から，平均値を予測することは難しいといえる.

次に Experiment 2 の標準偏差が大きくなる場合は，変化点が検出された．また Experiment 4 の標準偏差が小さくなる場合は，変化点が検出されなかった.

3.3　考察

第 3.2 節における実験の結果より，遅延分布の平均値の変化に対しては，変化点の検出が可能であることが分かった．そして標準偏差が大きくなる場合についても，これが可能である．また標準偏差が小さくなる Experiment 4 の場合は，スコア値に対して，しきい値を設ける今回の方法では，遅延分布の検知が難しいことが分かった.

しかし，遅延分布変化後のスコア値が実際に変動していることから，今回用いる方法と異なる方法で，この場合の予測可能性が示された．これは文献 4)では示されていない結果であるため，今後検討する余地がある.

次の章では，これを実際の DC モータ制御システムに応用し，リアルタイムで分布の変化に対応した制御システムを構築することで制御結果の改善がみられるのかを検証を行う.

4.　通信遅延時間の分布の変化に応じた制御に関する実験

4.1　ゲインスケジューリング制御

通信遅延時間の分布が変化する場合，適切なゲインに調整する必要がある．そのためには分布を知る必要がある．よってこれを行うために ChangeFinder を用いる．具体的には PC1 において，各サンプリング時間ごとにスコア値を出力し，これに変化が検出された場合，検出された以前 5 点の RTT(往復遅延時間 : Round Trip Time)の平均をとり RTT の値に対する適切なゲインにチューニングを行うことで，制御結果の改善を目指す．これは文献 4)と同様な方法である．分布の変化の検出は ChangeFinder のスコア値が 14 以上となった場合とする.

また，適切なゲインへのチューニングには，表 4 に示されるゲイン値のテーブルから選択することでチューニングを行った．表 4 中の L は往復遅延時間である．テーブルは RTT に対して IAE(Integral of Absolute value of Error)が最小になるよう作成した．IAE は(2)式で示される.

$$\int_0^\infty |\omega_E(t)|\,dt \qquad (2)$$

4.2 実験

以下に実験条件を示す. DC モータの目標回転速度は$\Omega_R(s)$=500 min^{-1}，そして PI 制御器のゲインの初期値は K_P=0.0019，K_I=0.018 とする．このゲイン値は遅延時間が 20 ms のときに，IAE が最小になるように設定されている．そして，制御時間を 4000 ms とし，1000 ms 時付近において，表 3 の条件で遅延分布の切り替えをし, DC モータの駆動実験を行う．

なお, ChangeFinder の各パラメータは第 3 章と同じ値を用い，そして今回行う実験でも, 図 1 における τ_b の遅延時間のみ変更し，τ_a=0 とした. また第 4 章の実験においても，制御システムの制約上，遅延分布を切り替える時刻付近で手動にて行うため，切り替え点が多少前後する．

4.3 実験結果

図 6 から 9 に実験結果を示す.

まず Experiment 1 では，後半の遅延分布の平均を 120 ms とした時に，本制御法を用いる場合，制御結果の改善が見られた．しかしそのほかでは, 逆にこれを用いる場合, 遅延分布変更時にモータの回転数が一度減少した後，目標回転数に収束する結果となった．

Experiment 2 では，本手法を用いる場合，2000 ms 付近ですでに収束しているのに対し，用いない場合は，その付近ではまだオーバーシュートおよびアンダーシュートが見られる．

Experiment 3 では, 後半の遅延分布の平均値を 80 ms とした場合に制御結果の改善が見られた．しかし，その他の場合は結果に変化は見られなかった．

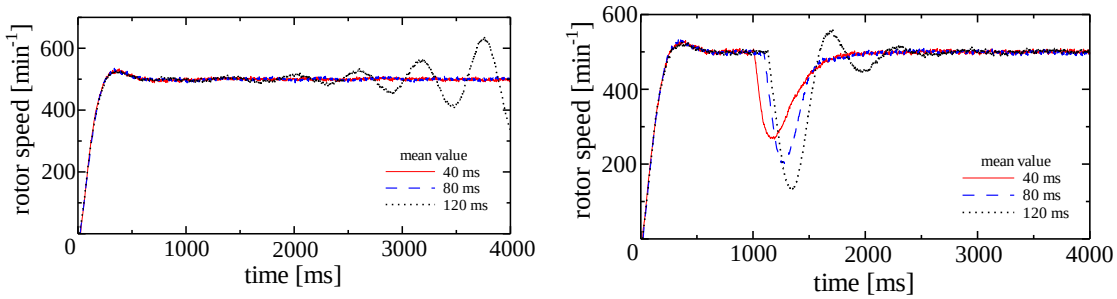
また Experiment 4 では，提案手法を用いない場合，応答が不安定であるのに対し，用いる場合は，目標回転速度に収束し，制御結果の改善が見られる．

4.4 考察

Experiment 1 において後半の遅延分布の平均を，実験条件の中では最も大きくした場合である 120 ms とした場合に，本制御法を用いない場合は，応答が振動的になっているが，本制御法を用いる場合，制御結果の改善が見られた．それ以外の場合については，手法を用いない場合では定常状態からの変動がないため，この場合逆に本手法を用いる必要がなかったと考えられる．このことから，より大きく遅延分布の平均値が変化する場合については，この制御法が特に有効であると考えられる．

次に Experiment 2 でも, 本制御法により，制御結果が改善されていると考えられる．ただし，制御結果が改善されたのは，標準偏差の変化の検出により，平均値に対応したゲインに調整されたためと考えられる．つまり，変化した標準偏差に対応したゲインに調整されたということではないと思われる．

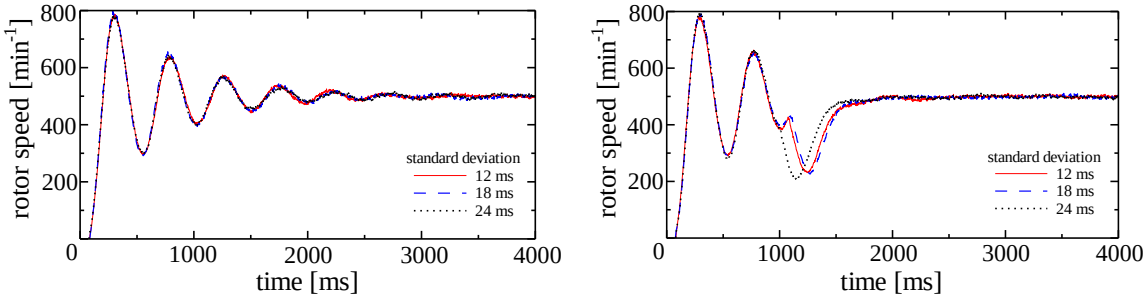
次に Experiment 3 では，後半の遅延分布の平均を 80 ms とした場合には制御結果の改善が見られた．しかしそれ以外の場合では改善が見られなかった．これは，後半の遅延分布の平均を小さく設定する実験であったため，初期ゲイン値からゲインの変更の必要がなかったためと考えられる．逆に 80 ms で改善が見られたのは, この平均値に



(a) without the proposed method　　(b) with the proposed method

図 6　Experiment 1 の結果

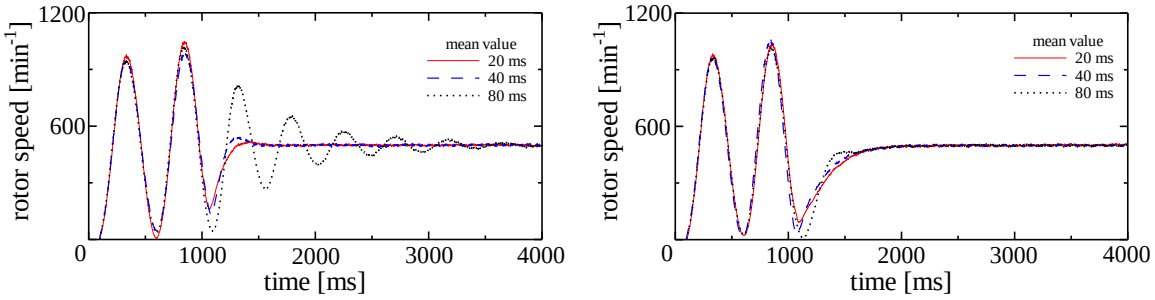
Fig. 6　Results of Experiment 1



(a) without the proposed method　　(b) with the proposed method

図 7　Experiment 2 の結果

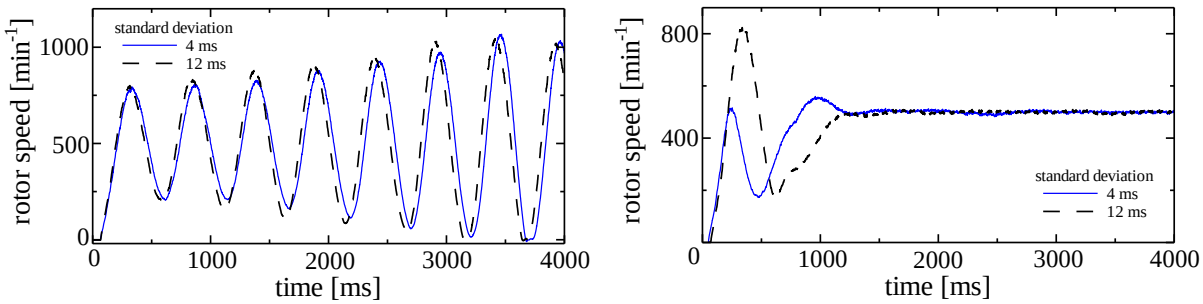
Fig. 7　Results of Experiment 2



(a) without the proposed method　　(b) with the proposed method

図 8　Experiment 3 の結果

Fig. 8　Results of Experiment 3



(a) without the proposed method　　(b) with the proposed method

図 9　Experiment 4 の結果

Fig. 9　Results of Experiment 4

対応したゲインに調整されたためと考えられる．

最後に Experiment 4 では，結果は改善しているように見られるが，これは図 4 の(d)で示されるように，分布の変化と関係なくスコア値が大きくなる点におけるスコア値により誤検出が起き，偶然スケジューリングがなされただけであると考えられる．よって本制御法で改善されたとは言えない．

5. おわりに

本研究では文献 4)で示された制御法を実際の IP ネットワークを介した制御システムに実装し，リアルタイムで制御を行うことで，その有用性についての検証を目的とした．

その結果，通信遅延時間分布の変化を ChangeFinder を用いて，リアルタイムで検出できることが確認された．また得られるスコア値から変化点を検出し，分布変化後の RTT からゲインスケジューリングを行う制御法も，平均値が大きく変化する場合は改善がみられた．また ChangeFinder による変化後の遅延分布の標準偏差の値の予測の可能性も示された．

今回の研究では ChangeFinder の各種パラメータを文献 4)と同じ値で用いたため，今後，他のパラメータに変更した場合についても考察する必要があると考えられる．

また本制御法のみでは結果が改善されない場合もあったため，この場合の改善策についても，検討の必要がある．

参考文献

1) R. A. Gupta and M.-Y. Chow: Networked Control System: Overview and Research Trends, IEEE Transaction on Industrial Electronics, **57**-7, 2527/2535 (2010)

2) Y. Tipsuwan and M.-Y. Chow: On the Gain Scheduling for Networked PI Controller Over IP Network, IEEE Transactions on Mechatronics, **9**-3, 491/498 (2004)

3) K. Matsuo, T. Miura and T. Taniguchi: A Speed Control of Small DC Motor through IP Network Considering Packet Loss, IEEJ Transactions on Electrical and Electronic Engineering, **2**-6, 657/659 (2007)

4) 高橋 勇人，松尾 健史，三浦 武，田島 克文：IP ネットワークを介した DC モータの通信遅延時間の変動に応じた制御に関する基礎的検討，計測自動制御学会東北支部第 297 回研究集会，297-6 (2015)

5) 山西 健司：データマイニングによる異常検知，共立出版，48/58 (2009)

6) netem: http://www.linuxfoundation.org/collaborate/workgroups/networking/netem