計測自動制御学会東北支部 第 259 回研究集会 (2010.7.9)
資料番号 259-6

IP ネットワークを介した小形 DC モータの遠隔制御法開発に関する検討
Consideration for Remote Control Method of a Small DC Motor through IP Network

○今野祐介*，松尾健史*，三浦　武*，田島克文*
○Yusuke Imano*，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)，ジッタバッファ(jitter buffer)

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 工学資源学研究科
松尾健史，Tel.：(018)889-2338，Fax.：(018)837-0406，E-mail：matsuo@ipc.akita-u.ac.jp

1. はじめに

現在，パーソナルコンピュータや携帯電話などの端末から情報を得る手段の一つにインターネットがある．現在，インターネットすなわち IP ネットワークは世界に広がる大きなネットワークを形成しており，広く普及している．近年，企業が国外にプラントを建設する機会も多い社会背景を踏まえると，IP ネットワークの産業応用に対する需要があると考えられる．IP ネットワークを用いる利点として，広く普及しているネットワークを利用し，既存の通信ネットワークを用いることができるため，新たな通信設備の投資も必要なく，低コストでのシステム構築が可能になる．また，離れた地点から対象を制御することが可能になる．

しかし，IP ネットワークを制御に用いる場合，遅延時間や遅延時間の揺らぎ(ジッタ)，パケット損失は制御性能を低下させる原因となる．そのため，これまで様々な手段で対処がされてきた．文献 1)では，遅延時間の揺らぎの影響を抑える手法として，受信するデータを蓄えるバッファ(ジッタバッファ)を用いる手法が提案されている．このジッタバッファは現在のサンプリングデータの他に，過去のサンプリングデータを格納し，過去のデータから処理していくことで，遅延時間の揺らぎの影響を抑えることができる．また，文献 2)では，受信にジッタバッファを用い，送信に現在のサンプリングデータに加え，過去のサンプリングデータもまとめて送信する手法(以下冗長伝送と呼ぶ)を用いパケット損失の影響を抑える手法が提案されている．

そこで，本研究では IP ネットワークシス

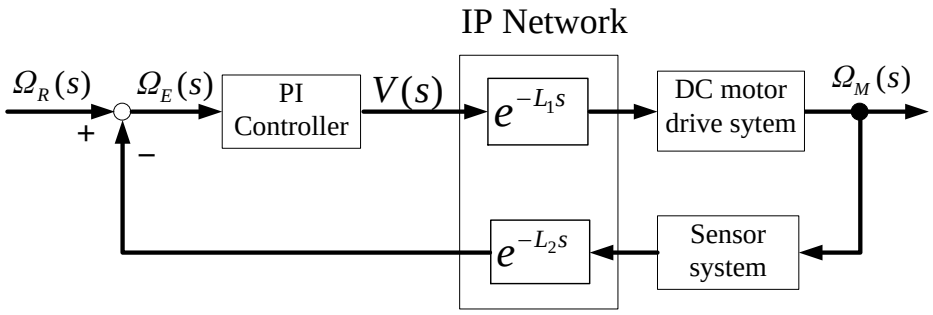


図 1 IP ネットワークを介した DC モータ遠隔制御システムのブロック線図
Fig.1 Block diagram of remote control system for a DC motor through IP network.

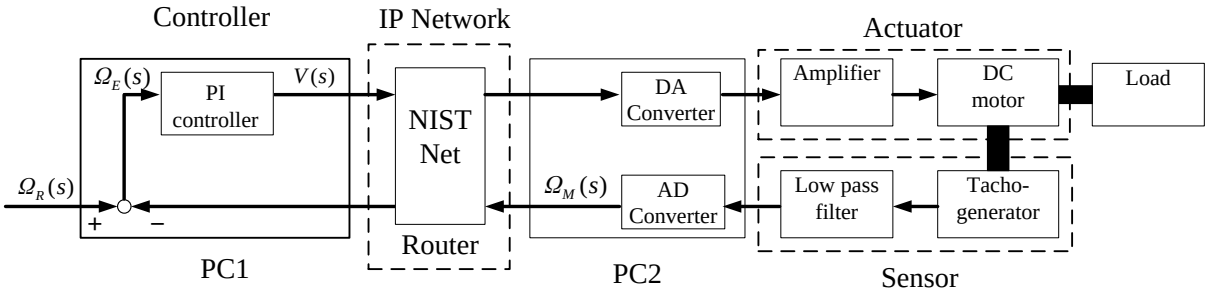


図 2 IP ネットワークを介した DC モータ遠隔制御システムの構成図
Fig.2 Configuration of remote control system for a DC motor through IP network.

テムを構築し，遅延時間に揺らぎが存在する環境下において DC モータの速度制御を行い，ステップ応答のばらつきを抑えられないか検討を行った．本研究は 2 つの実験を行い，実験 1 では文献 1)の受信方法を利用し，むだ時間をなるべく小さくしてステップ応答のばらつきを抑えられないか検討を行い，実験 2 では文献 2)の考え方を基に，遅延時間の揺らぎによる影響に対しても，冗長伝送により抑えることができるのではないかと考え，実験を行った．

2. 実験システム

＜2・1＞実験システムの構成

本研究では，IP ネットワークを介して DC モータを駆動するシステムを用いる．本実験システムのブロック線図を図 1 に示し，実験システムの構成図を図 2 に示す．ここで，$\Omega_R(s)$ は目標回転速度[min^{-1}]，$\Omega_M(s)$ は実際の DC モータの回転速度[min^{-1}]，$\Omega_E(s)$ ($\Omega_E(s) = \Omega_R(s) - \Omega_M(s)$)は目標回転速度と実際のモータの回転速度との偏差[min^{-1}]，$V(s)$ は操作量(DC モータへの印加電圧の指令値)[V]，ネットワーク部の L_1，L_2 は通信遅延によるむだ時間[s]を示している．むだ時間は，遅延時間の揺らぎが入ると値が変化するため定数ではない．

本研究では，Personal Computer 1(以下 PC1)は PI 制御を行うコントローラ側とし，Personal Computer 2(以下 PC2)は DC モータへの DA 変換，タコジェネレータからの AD 変換を行うコンピュータとして使用している．

IP ネットワーク部には，ルータのネットワークエミュレータである NIST Net [3)]がインストールされた Personal Computer が挿入されている．実際の IP ネットワークを用いる場合，通信遅延や，物理的な距離，ルータの性能，ネットワークの混雑度，通信相手のコンピュータの負荷などにより起こる遅延時間の揺らぎ，そしてルータがデータを転送しき

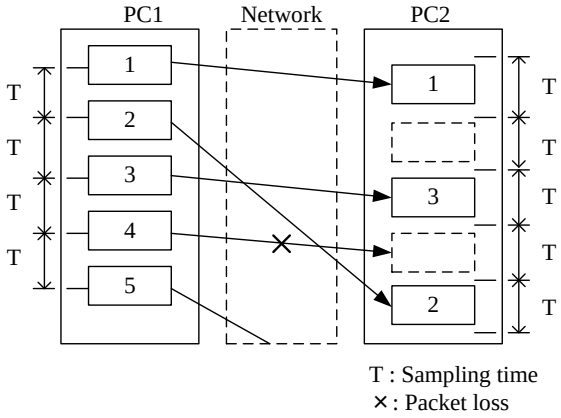


図 3 ジッタバッファを用いない通信
Fig.3 Transfer without buffer.

表 1 DC モータの仕様
Table 1 Specification of a DC motor.

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

れなくなったときや，輻輳時などに発生するパケット損失 [4]など様々な要因を考えなくてはならない．そこで本実験ではネットワークエミュレータにより実際のネットワークを模擬して実験を行った．本実験で使用した NIST Net では通信遅延時間[ms]とその揺らぎの標準偏差[ms], パケット損失率[%]などのネットワーク環境を設定し，模擬することができる．

本実験では通信プロトコルである TCP (Transmission Control Protocol) / IP(Internet Protocol)と比べ，リアルタイム性に優れる UDP (User Datagram Protocol)を用いた．

PI 制御器の伝達関数は(1)式となる．

$$G_{PI}(s) = K_P + \frac{K_I}{s} \qquad (1)$$

K_P および K_I はそれぞれ比例ゲインと積分ゲインを表す．本実験では K_P =0.0007，K_I =0.0025 として実験を行った．

DC モータの回転速度は，タコジェネレータを介して電圧値として PC2 に取得される．その情報は IP ネットワークを介してコントローラ側である PC1 のバッファに格納される．PC1 では，回転速度の目標値と現在速度の目標値の偏差を基に PI 演算を行い，その

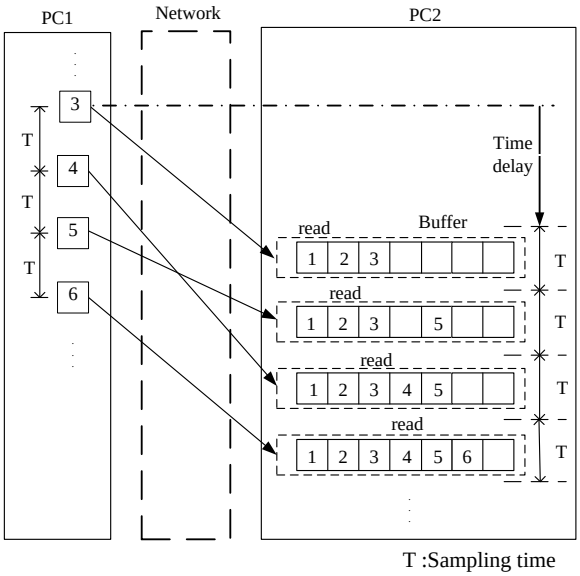


図 4 ジッタバッファを用いた通信 [1)]
Fig.4 Transfer with jitter buffer [1)].

情報は IP ネットワークを介して PC2 のバッファに格納される．PC2 では受信した操作量を，DA 変換器によってアナログ電圧に変換し，増幅器で 5 倍に増幅された電圧が DC モータに印加される．

今回のシステム(図 2)において制御対象とした DC モータは山洋電気社製 DC サーボモータ R301T-011 である．その仕様を表 1 に示す．タコジェネレータは 3V/1000min^{-1} のものを使用した．PC2 には，DA・AD 変換器を取り付けており，PC1，PC2 ともにサンプリング時間は 1ms である．受信は 0.5ms 経過しても届かない場合は，パケット損失として扱い，前の回転速度(印加電圧指令)のデータを用いて制御している．慣性負荷は 2.5×10^{-5} N・m・s^2/rad のものを装着した．

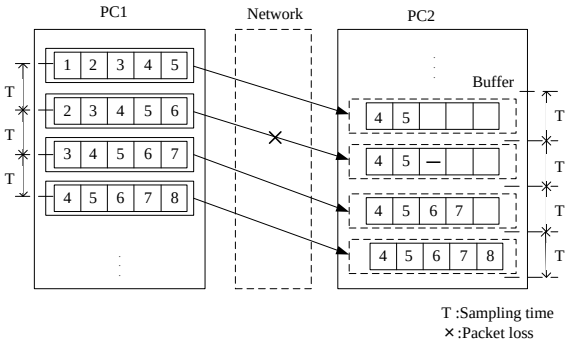


図 5 冗長伝送を用いた通信 [2)]
Fig.5 Transfer using redundant transmission[2)].

＜2・2＞受信方式

遅延時間に揺らぎが存在すると，遅延時間が変動するため，パケットを送信してから受信されるまでの時間がパケットごとに異なる．文献 1)では受信したパケットをバッファリングし，バッファ(ジッタバッファ)にデータを格納することで揺らぎの影響を抑える手法を提案している．ジッタバッファを用いない受信では，遅延時間の揺らぎの影響を受ける(図 3).

通信相手は送信するデータに番号をつけることで，受信の際ジッタバッファには送信された順番にデータを溜め込んでいくことができる．そして現在の値ではなく，過去のデータをもとにデータを処理していく．どの程度前のデータから処理していくのかは設定でき，その値を本研究ではバッファ数と呼ぶことにする．バッファ数が 3 の場合は，6 個のデータをバッファに格納している状態を例に取ると，4 番目のデータから読み込み，処理を行うことになる(図 4)．本実験ではサンプリング時間を 1ms としているため，バッファ数が 3 の場合は 3ms 前のデータから読み込んで処理を行うことになる．これによりバッファ数の時間の分，遅延時間の揺らぎによってパケットが受信される順番の入れ替わりを防ぐことができる．

文献 1)では，揺らぎのある環境下で正弦波信号を送信し，バッファ数を大きく設定することで，むだ時間を一定にし，応答の再現性を良くできるといった報告がされている．しかし制御において，バッファは過去のデータを処理するため，その時間はむだ時間となる．よって，バッファ数は小さいほうが好ましい．そこで本実験では，DC モータの速度制御を行う際，バッファ数を小さくしてステップ応答のばらつきを抑えることができるか検討した．

＜2・3＞送信方式

遅延時間に揺らぎが存在する場合，遅延時間が大きくなるパケットに関しては，サンプリング時間内に受信されない場合データが届かないためパケット損失となってしまう場合がある．そこで文献 2)では，図 5 に示すように，受信にジッタバッファを用い，送信の際に複数のサンプリングデータをまとめて送信する手法(冗長伝送)が提案されている．パケット損失が発生する環境下において，失われたデータを後の送信により補うことが可能になるため，一つ一つデータを送るのに対し，パケット損失に強い伝送方式となる．本実験では，遅延時間に揺らぎのある環境下で冗長伝送を用い，遅延時間の揺らぎによって発生するパケット損失のデータを補うことで揺らぎの影響を抑えることができるのではないかと考え，実験では冗長伝送を用いた DC モータの速度制御を行った．

3. 実験

遅延時間の揺らぎの影響を抑えるために二つの実験を行った．

実験 1：受信にジッタバッファのみを用い，
　　バッファ数を変えて，
　　DC モータの速度制御を行った．
実験 2：受信にジッタバッファを用い，
　　送信に冗長伝送を用いて
　　バッファ数を変えて，
　　DC モータの速度制御を行った．

本実験の環境として，ネットワークエミュレータ NIST Net の設定により遅延時間を L_1=L_2=100ms とし，遅延時間の揺らぎの標準偏差を 20ms とした． PC1 で取得した RTT は遅延時間の揺らぎにより毎回異なる．そのうちの一つの RTT の分布を図 6 に示す．

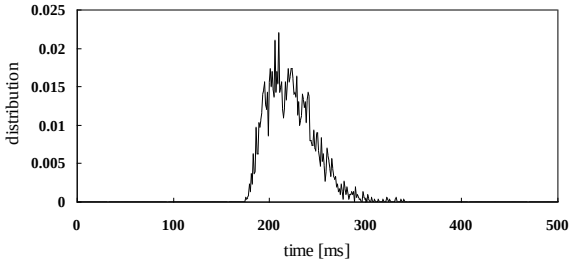


図 6　RTT の分布
Fig.6 Distribution of RTT.

実験 1 ジッタバッファのみを用いた PI 制御

遅延時間に揺らぎが存在する環境下では，ジッタバッファによって揺らぎによる順番の入れ替わりを防ぐことができる．しかし，バッファ数を大きくすると，その分だけ過去のデータを処理することとなり，むだ時間も大きくなってしまう．このためバッファ数は小さく設定することが好ましい．

実験では，DC モータの目標回転速度を 1500min^{-1} として，バッファ数を 0，10，20，

30 と変えてステップ応答を取得し, 遅延時間の揺らぎの影響を抑えることができるか検討した.

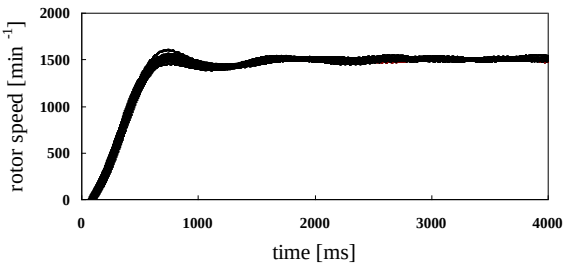
以下には(a)バッファ数 0 の場合, (b)バッファ数 10 の場合，(c)バッファ数 20 の場合，(d)バッファ数 30 の場合として，図 7 に 30 個のステップ応答波形，図 8 にオーバーシュート量が最も大きい max と最も小さい min のステップ応答波形，図 9 にオーバーシュート部分をそれぞれ示す.

実験 2 ジッタバッファと冗長伝送を用いた PI 制御

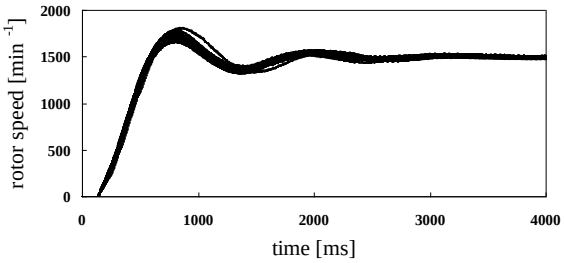
遅延時間に揺らぎが発生している環境下において，冗長伝送を用いることで，遅延時間が大きいパケットのデータを，比較的遅延時間が短いパケットが補うことで，揺らぎの影響を抑えることができるか検討した.

実験では，DC モータの目標回転速度を 1500min^{-1} とし, ジッタバッファと冗長伝送を用い，バッファ数を 10，20，30 と変えてステップ応答を取得した.

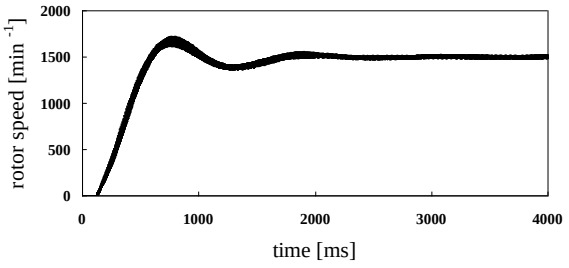
以下には(a)バッファ数 10 の場合，(b)バッファ数 20 の場合，(c)バッファ数 30 の場合として，図 10 に 30 個のステップ応答波形，図 11 にオーバーシュート量が最も大きい max と最も小さい min のステップ応答波形, 図 12 にオーバーシュート部分をそれぞれ示す.



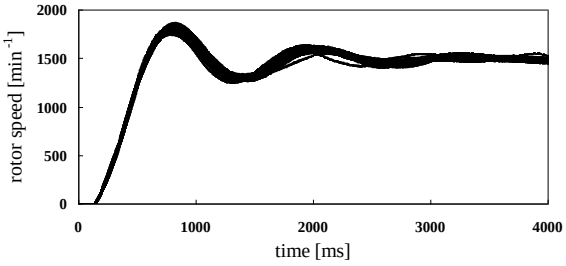
(a) バッファ数 0
(a) Buffer number 0.



(c)バッファ数 20
(c) Buffer number 20.

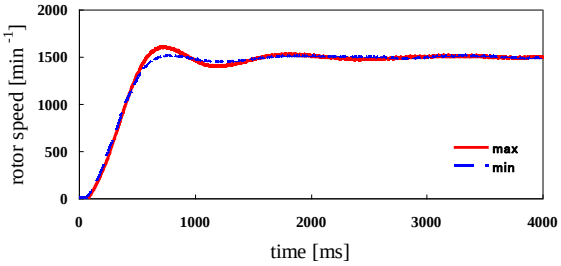


(b) バッファ数 10
(b) Buffer number 10.

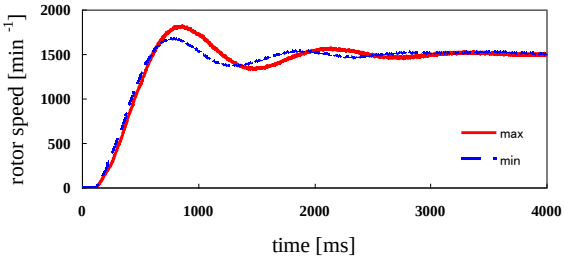


(d)バッファ数 30
(d) Buffer number 30.

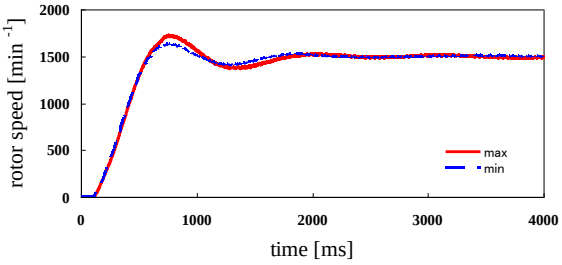
図 7 ジッタバッファを用いたステップ応答波形
Fig.7 Step responses with jitter buffer.



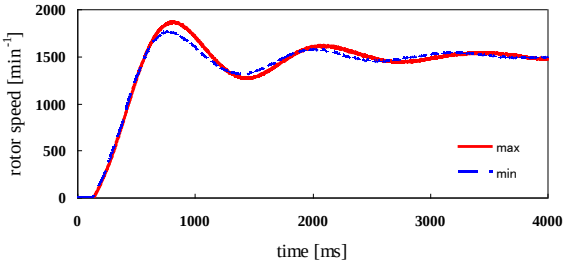
(a) バッファ数 0
(a) Buffer number 0.



(c)バッファ数 20
(c) Buffer number 20.

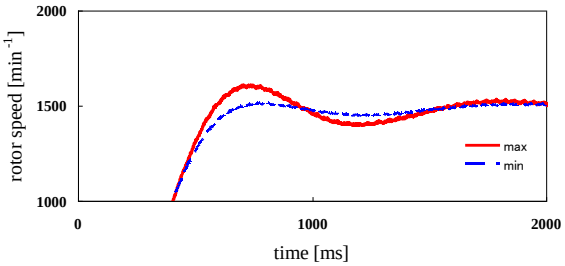


(b) バッファ数 10
(b) Buffer number 10.

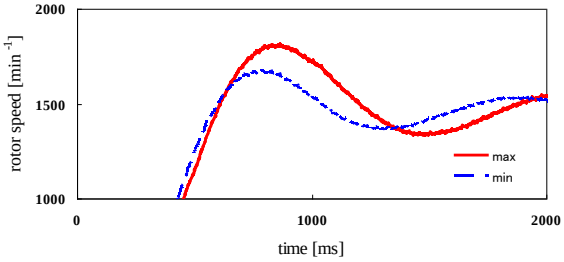


(d)バッファ数 30
(d) Buffer number 30.

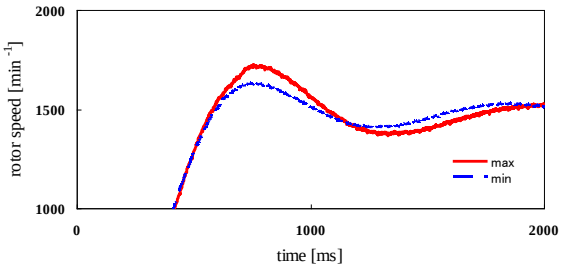
図 8 max と min のステップ応答波形
Fig.8 Step responses with the maximum overshoot and those with the minimum one.



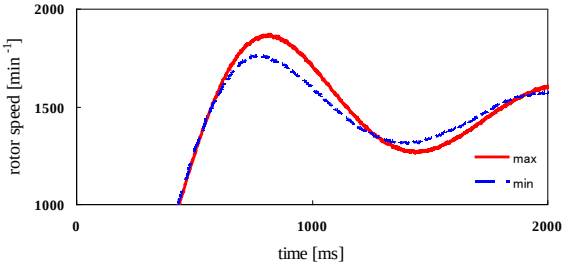
(a) バッファ数 0
(a) Buffer number 0.



(c)バッファ数 20
(c) Buffer number 20.

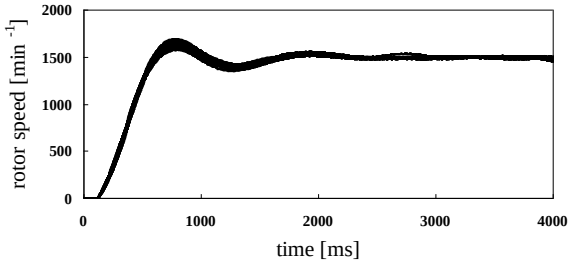


(b) バッファ数 10
(b) Buffer number 10.

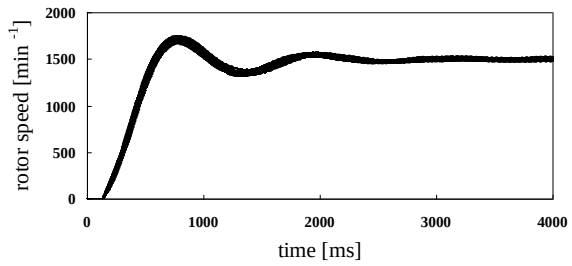


(d)バッファ数 30
(d) Buffer number 30.

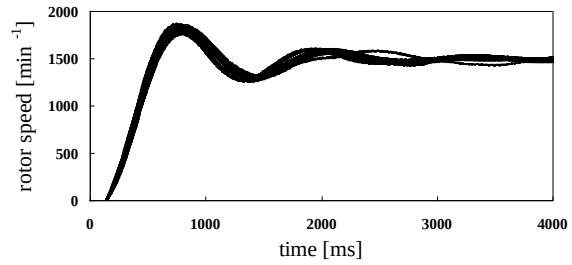
図 9 オーバーシュート部分
Fig.9 Overshoot part.



(a) バッファ数 10
(a) Buffer number 10.

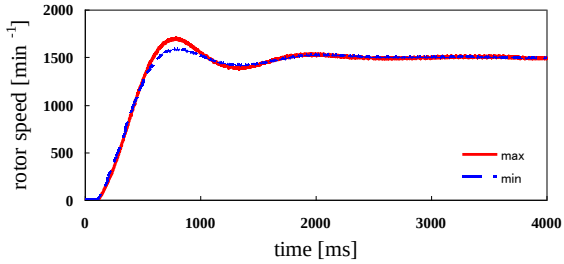


(b) バッファ数 20
(b) Buffer number 20.

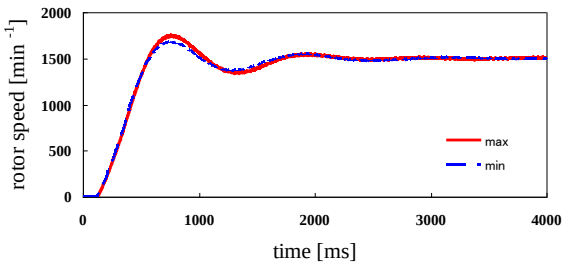


(c) バッファ数 30
(c) Buffer number 30.

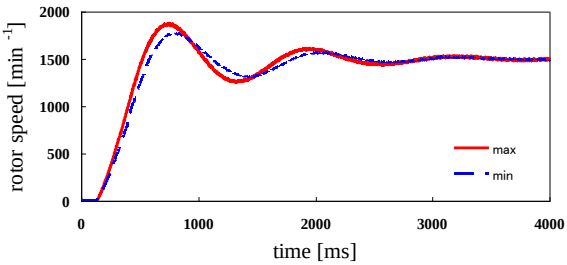
図 10 ジッタバッファと冗長伝送を用いたステップ応答波形
Fig.10 Step responses with jitter buffer using redundant transmission.



(a) バッファ数 10
(a) Buffer number 10.



(b) バッファ数 20
(b) Buffer number 20.



(c) バッファ数 30
(c) Buffer number 30.

図 11 ジッタバッファと冗長伝送を用いた max と min のステップ応答波形
Fig.11 Step responses with jitter buffer using redundant transmission.

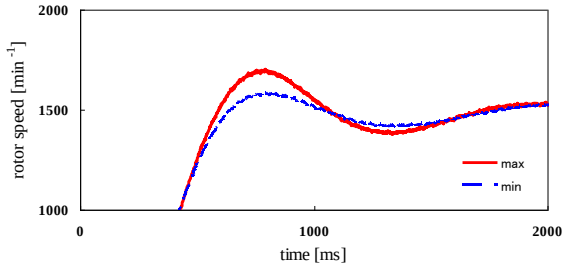
4. 考察

＜4・1＞実験 1 の考察

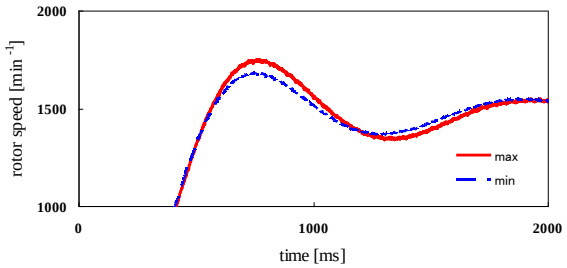
実験 1 では，ジッタバッファのバッファ数を変えてステップ応答波形を取得した．図 7 より，バッファ数 0 のときと比べ受信にジッタバッファを設けることで，立ち上がりのばらつきを抑えることができていることが確認できる．また，バッファ数 0 と比べ，バッファ数 10 では，ステップ応答のばらつきが小さくなっていることが確認できるが，バッファ数が 20，30 のときはステップ応答のばらつきを抑えることができていないことが分かる．これは，ジッタバッファによるむだ時間が大きくなり，オーバーシュート量が大きくなったと考えられる．

＜4・2＞実験 2 の考察

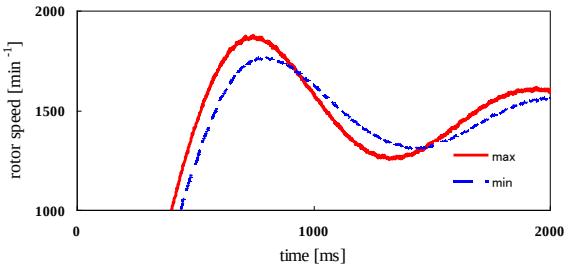
実験 2 では，受信にジッタバッファを用い，送信に冗長伝送を用いてステップ応答を取得した．送信に冗長伝送を用いることで，受信の際に遅延時間の揺らぎの影響を抑えることができると考えられるが，図 7 と図 10 を比較すると，バッファ数 20 においては，冗長伝送を用いた方はステップ応答波形のばらつきを抑えられていることが確認できる．しかし，バッファ数 10，30 では，冗長伝送を用いた場合と比較しても大きな違いは見られなかった．



(a) バッファ数 10
(a) Buffer number 10.



(b) バッファ数 20
(b) Buffer number 20.



(c) バッファ数 30
(c) Buffer number 30.

図 12 ジッタバッファと冗長伝送を用いたオーバーシュート部分
Fig.12 Overshoot part of step responses with jitter buffer using redundant transmission.

5. おわりに

今回，ジッタバッファのみ用いた PI 制御を行った結果，バッファ数が小さいときはステップ応答のばらつきが抑えられたが，バッファ数を大きくするとむだ時間が大きくなる影響でステップ応答のばらつきが大きくなることを確認した．また，ジッタバッファと冗長伝送を用いる PI 制御を行った結果，一部ステップ応答のばらつきを抑えることが確認できたが，一部を除きステップ応答のばらつきに大きな違いが見られなかった．

謝辞

本研究に関する貴重なご意見を頂いた秋田大学教授 故谷口敏幸先生に哀悼の意と共に感謝の意を表する．

参考文献

1) 加藤 敦，西 宏章，大西公平：ジッタバッファを用いたネットワークバイラテラル制御システム，電学論 D，**126**-12，1737/1738 (2006)

2) Kenshi Matsuo，Takeshi Miura，Toshiyuki Taniguchi：A Speed Control Method of Small DC Motor Through IP Network Considering Packet Loss，IEEJ Transactions on Electrical and Electronic Engineering，**2**-6，657/659 (2007)

3) NIST Net：
http://www-x.antd.nist.gov/nistnet/

4) 村山公保：TCP/IP コンピューティング入門，96/101，オーム社 (1999)