計測自動制御学会東北支部 第 291 回研究集会 (2014.10.23)
資料番号 291-6

IP ネットワークを介した小形 DC モータ 速度制御システムにおける送受信に関する検討

A Study for Transmitting and Receiving in a Small DC Motor Speed Control System through IP network

○高橋朋広*，松尾健史*，三浦　武*，田島克文*
○Tomohiro Takahashi*, 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)，通信時間(time delay)

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

近年，パーソナルコンピュータやスマートフォンの普及により，インターネット，すなわち IP ネットワークは, 我々の生活に欠かせなくなってきている．

これを利用した応用に IP ネットワークの産業応用があり, その 1 つに IP ネットワークを介した遠隔制御が挙げられる．この制御システムを構築するとき，従来のシステムのように専用回線を準備することなく，既存の IP ネットワークにコントローラ，センサ，アクチュエータ等を接続するだけで容易にかつ安価に構築が可能となる利点がある．

しかし，これを制御する場合，IP ネットワークの通信遅延時間およびその揺らぎが原因で，制御性能が劣化する問題がある[1)]．さらに，揺らぎについては，送信した信号の順番と到着する信号の順番が入れ替わる場合[2)]もあり，これも劣化の原因となる．

そのため，これに対するさまざまな制御法が提案されているが，そのうち送受信法で改善する方法の 1 つとして，アクチュエータの操作量を加えるときに，タイムスタンプを参照することで最新のデータを使用する手法[2)]がある．

さて，ネットワークを介さない従来の制御システムにおいて，コントローラ，センサ，アクチュエータは，一般にすべてサンプリング時間毎，すなわち，タイムドリブン法で制

御する．このタイムドリブン法でネットワークを介した制御システムを構築する場合，それぞれサンプリング時間で制御信号の送受信を行う．このとき，通信遅延時間によって制御信号を受信する間隔が異なる場合があるため，制御信号がサンプリング時間内に受信されない場合がある．そこで，例えば文献 3)のように，ネットワークを介した場合，制御信号を受信するまでコントローラとアクチュエータを待機状態とし，受信に合わせて動作させるイベントドリブン法で制御するシステム構成もある．

本研究で構築する遠隔制御システムは，文献 4)のようにコントローラとアクチュエータおよびセンサが IP ネットワークを介して接続され，タイムドリブン法で制御されるシステムを想定する．しかし，このシステムにおいて，アクチュエータおよびセンサはタイムドリブン法で動作させる必要があるが，コントローラはタイムドリブン法，イベントドリブン法どちらを用いてもシステムを構築することができる．

このとき，コントローラの動作をタイムドリブン法あるいはイベントドリブン法によってシステムを構成し制御した場合，それぞれの制御性能を実システム上で実装して比較したい．さらに，文献 2)の手法を参考にして，この文献で適用しているアクチュエータ部ではなく，コントローラの受信時に入れ替わりに対する手法を適用し，その性能についても検討したい．

そこで本研究では，システムのコントローラ部に注目して，パケットの入れ替わりに対してタイムスタンプにより最新データを使用する手法の性能，送受信をタイムドリブン法とイベントドリブン法で行った場合の性能比較，およびそれらを組み合わせて使用した場合の性能を代表的なサーボモータである小形 DC モータを用いた実際の制御システムにおいて検討する．具体的には，上記の手法を用いて速度制御実験を行い，その結果を考察する．なお，アクチュエータ部においては特別な手法を用いないものとする．

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小形 DC モータを駆動する実験システムを用いる．実験システムの構成図を図 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$\omega_e(t) = \omega_r(t) - \omega_m(t)$ [min^{-1}], $v(t)$は DC モータへの印加電圧[V]である．また，τ_a, τ_bは IP ネットワークを介した際の通信遅延時間である．

本実験システムは IP ネットワークを介して PC1 と PC2 が接続されているシステムを想定している．ここで，PC1 は PI 制御を行うコントローラの役割を果たすパーソナルコンピュータ，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パーソナルコンピュータである．また，PC1，PC2 は共にパケットの送受信を行う．IP ネットワーク部には Router PC があり, PC1 と PC2 の通信時データを中継する．このときの，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[5)]が導入されている．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って通信遅延時間やその揺らぎが変動するが，それらを任意に発生させることは困

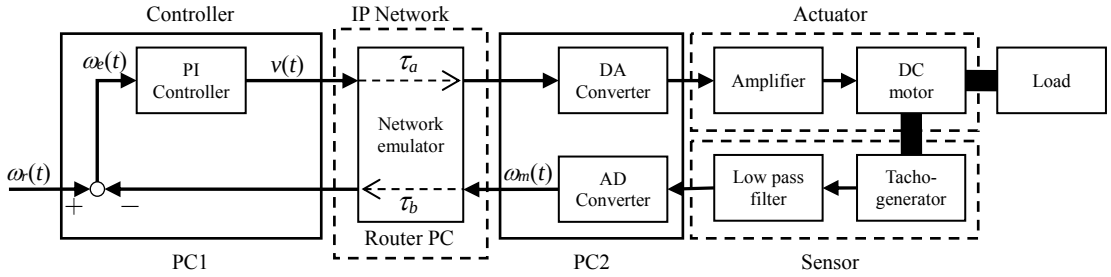


図 1 IP ネットワークを介した DC モータ遠隔制御システムの構成図

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

難である．そこで本研究では netem によって通信遅延時間[ms]とその揺らぎ[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．

次に，実験システムの流れを示す．PC1 で目標回転速度と実際のモータの回転速度との偏差を PI 制御器に入力することで操作量である印加電圧が計算される．印加電圧は PC1 から送信され，Router PC による IP ネットワーク部を介して PC2 で受信される．PC2 で受信された印加電圧は DA 変換器によってアナログ信号へ変換され，電圧増幅器を介して DC モータに印加し駆動させる．DC モータと慣性負荷およびタコジェネレータが機械的に接続されており，タコジェネレータによって検出された回転速度はローパスフィルタ, AD 変換器を介して PC2 で取得される．取得された回転速度は再びネットワークを介して PC1 へフィードバックされる．

制御対象には山洋電機社製 DC サーボモータ R301T-011 を使用する．その仕様を表 1 に示す．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．

2.2 送受信方法

IP ネットワークを介した制御では図 3 に示すように通信遅延時間とその揺らぎの影響を受ける．そこで本節では，コントローラ部をタイムドリブン法，イベントドリブン法として送受信を行う手法および，タイムスタンプを用いたパケットの入れ替わりに対する受信法について述べる．

2.2.1 タイムドリブン法およびイベントドリブン法による送受信法

例えば，文献 3)では制御側である PC1 のサンプリング時間を DC モータの駆動側である PC2 と同じとして実験を行っている．しかし，図 3 のように，通信時間遅延の揺らぎによって，パケットが到着するまでの時間はパケット毎に異なる．そのため，あるサンプリング時間 T の時間内で受信されない場合がある．

そこで本研究ではコントローラの動作を一定のサンプリング時間毎に制御を行うタイムドリブン法と，パケットを受信した毎に制御を行うイベントドリブン法をそれぞれ用いる．

図 4(a)は PC1 がタイムドリブン法で動作する送受信である．タイムドリブン法の場合，PC1 はサンプリング時間毎に送受信および PI 制御の計算を行う．PI 制御の操作量は式(1)から求められる．

$$v(kT) = k_p\omega_e(kT) + k_i\sum_{s=1}^{k}\omega_e(sT)\cdot T \qquad (1)$$

ここで，k_p は比例ゲイン，k_i は積分ゲイン，である．

表 1　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}

一方，図 4(b)は PC1 がイベントドリブン法で動作する送受信である．イベントドリブン法の場合, PC1 が時刻 t_0 で制御を開始すると，パケットが受信されるまで待機状態となる．パケットが受信されると，前回の受信との受信間隔で制御が行われる．例えば，時刻 t_3 で受信した場合，受信間隔は前回の受信時刻との差，すなわち t_3–t_2 となる．受信間隔は，揺らぎによって受信の間隔が異なるため，一定の値ではない．よって式(2)から操作量が計算される．

$$v(t_i) = k_p \omega_e(t_i) + k_i \sum_{s=1}^{j} \omega_e(t_s) \cdot (t_s - t_{s-1}) \quad (2)$$

なお，j は受信をした回数である．式(2)の計算後，制御信号を送信し，再び次の受信まで待機状態となる．

2.2.2　タイムスタンプによるパケットの入れ替わりに対する送受信法

ここでは，文献 2)の手法を参考にしたパケットの入れ替わりに対する送受信法を説明する．文献 2)ではアクチュエータ部で行われている手法を，本研究ではコントローラ部で適用する．以下，本研究で用いる手法を説明する. コントローラやアクチュエータが IP ネットワークを介して通信を行う場合，通信遅延時間が揺らぎによって変動するため，送信したパケットが受信されるまでの時間がパケットごとに異なる．さらにその際に，図 3 のようにパケットの入れ替わりが発生する場合があり，制御時に過去のデータが使われてしまう．そこで常に最新のデータを用いるためにタイムスタンプが利用される．図 5 にタイムスタンプを用いた送受信について示す.

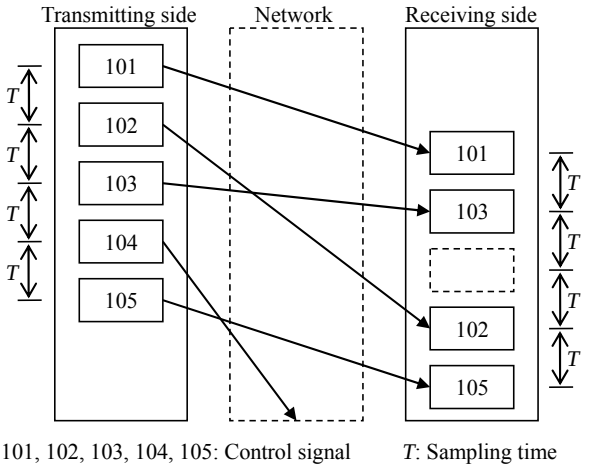
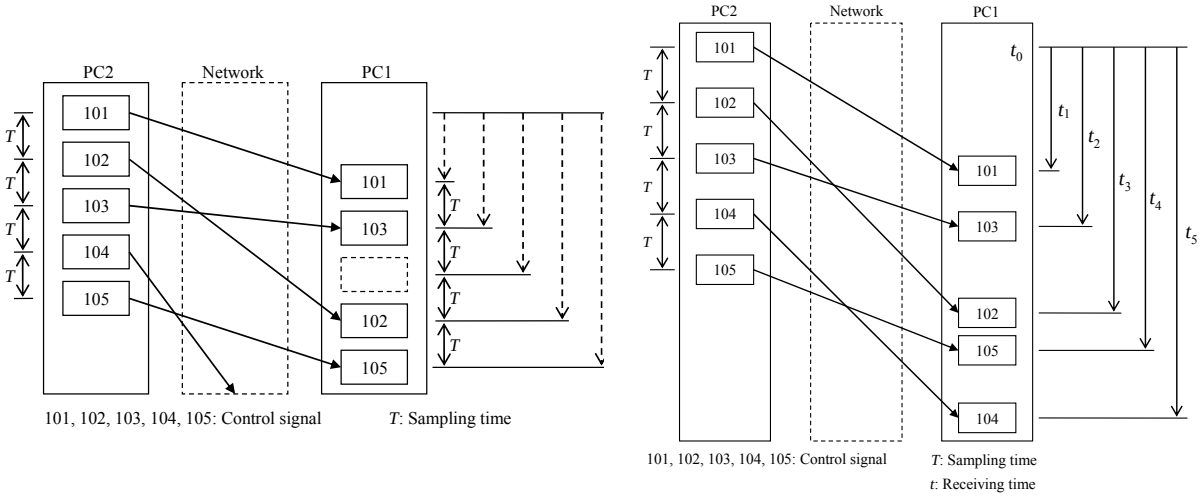


図 2 IP ネットワークを介した送受信

Fig.2 Transmitting and receiving through IP network

図 5(a)はタイムスタンプをタイムドリブン法で行う場合である．送信側は制御信号を送信する際に，制御信号にタイムスタンプを付けて送信を行う．受信側は受信した制御信号のタイムスタンプの番号と，直前のタイムスタンプの番号とを比較し，過去のデータを破棄することで，常にタイムスタンプの新しいデータを用いて制御を行うことができる．過去のデータを破棄する場合は，その直前のデータを再度使用して制御を行っている．

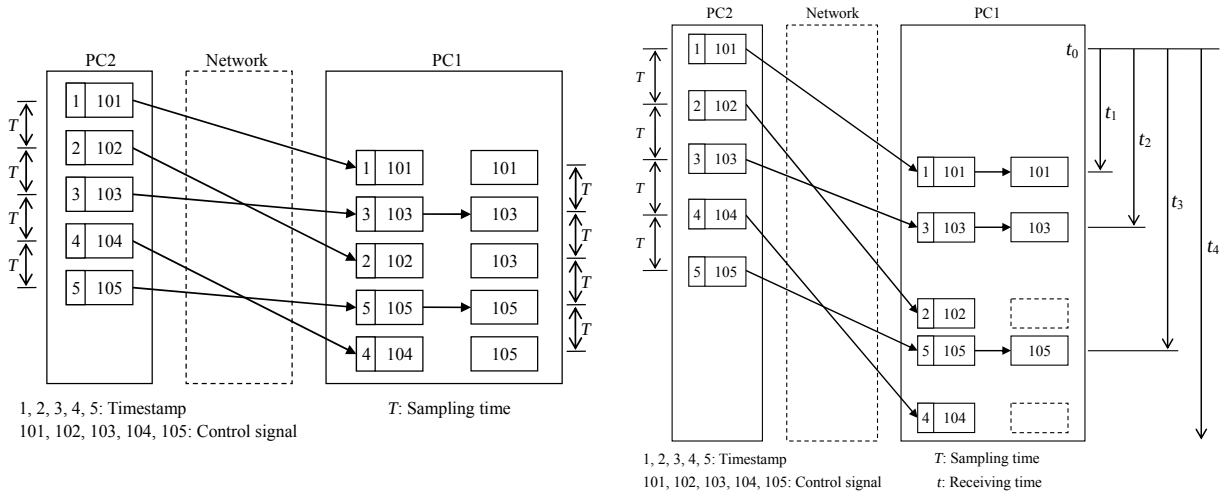
また本研究ではタイムスタンプによるパケットの入れ替わりに対する送受信法をイベントドリブン法と組み合わせた場合についても実験を行う．その場合の送受信を図 5(b)に示す. イベントドリブン法で動作するため, PC1 は受信が行われるまで待機状態となる．しかし，この場合ではタイムスタンプにより過去のデータは破棄されるため，PI 制御の計算および送受信が行われるのは最新のデータを受信したときのみとなる.



(a) Time-driven　　(b) Event-driven

図 3 タイムドリブン法およびイベントドリブン法による送受信

Fig.3 Transmitting and receiving with the time-driven method and event-driven method.



(a) Timestamp＋Time-driven　　(b) Timestamp＋Event-driven

図 4 タイムスタンプによる送受信

Fig.4 Transmitting and receiving with timestamps.

3. 実験

2.2 節で述べた送受信法を用いて実験を行い，送受信によって DC モータの制御を改善することが可能か検討する．具体的には，本実験では PC1 を，従来の制御法を単純にネットワークを介した制御に拡張した場合の送受信法であるタイムドリブン法とタイムスタンプによる入れ替わりに対する手法なしの場合(送受信法(a))と次の(b)-(d)の送受信法で実験を行い，送受信法(a)と比較することで，(b)-(d)の送受信法の制御性能について検討する．PC2 はタイムドリブン法で動作し，送受信法による性能を比較するために，PC2 では送受信および，DA/AD 変換以外を行わない．また，タイムドリブン法で動作させる場合のサンプ

表 2 実験条件

Table 2 Experiment Condition.

Reference speed	1500 min^{-1}
Time delay (τ_a=τ_b)	50 ms
Jitter	10, 30 ms

リング時間は 1ms とする.

(a) タイムドリブン＋タイムスタンプなし

(b) タイムドリブン＋タイムスタンプあり

(c) イベントドリブン＋タイムスタンプなし

(d)イベントドリブン＋タイムスタンプあり

実験条件を表 2 に示す．揺らぎによる影響を確認するために, 実験条件毎に 20 回の実験を行う． また, 比例ゲインおよび積分ゲインは(a)の送受信を使い, τ_a, τ_b=50 ms 揺らぎ 0 ms の環境で試行錯誤的に k_p=0.0017，k_i=0.0057 と設定した．実験結果を図 5 に示す．各図には比較として揺らぎ 0ms で取得したステップ応答も示す.

4. 考察

(a)-(d)の送受信法を用いてステップ応答の取得を行った．図 5(A)で示される揺らぎが 10ms の場合，(a)-(d)のステップ応答に大きな変化は見られず，送受信法による違いは見られなかった．これは揺らぎによる影響が少ないことが原因と考えられる.

一方，図 5(B)に示される，揺らぎが 30ms の場合では，送受信法ごとにステップ応答に変化が見られた．以下に(b)-(d)の送受信法に関して，送受信による手法を用いていない送受信法(a)と比較してその性能について考察する.

(b)の場合では同じ実験でのステップ応答(a)と比較すると，ステップ応答オーバーシュートが全体的に小さくなる傾向が見られた．また，ステップ応答のばらつきに関しても多少の減少が見られた.

(c)の場合ではステップ応答(a)と比較して多少ばらつきの減少が見られた．オーバーシュートの値に関しては他の送受信法のように小さくなる傾向はなかった.

また(d)の場合では，ばらつきが大幅に減少する傾向がみられた．ばらつきが減少する一方で，揺らぎが大きくなるにつれて，揺らぎ 0ms で取得したステップ応答と比較して，目標回転速度に達する前，オーバーシュート以下のピークが発生し，その後，目標回転速度に近づくという傾向があることがわかった.

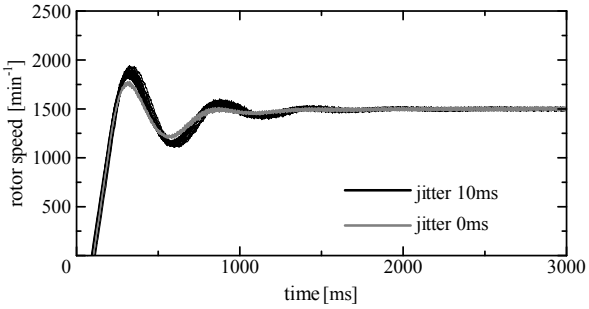
これは，(d)の送受信法ではパケットを受信した時のみ PI 制御の計算を行っているが, 揺らぎ 0ms の場合と比べて PI 制御で計算される操作量が減少しているため，結果オーバーシュートの減少が発生した.

5. おわりに

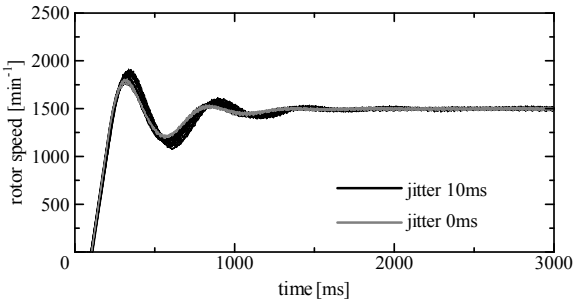
今回，コントローラにタイムスタンプによるパケットの入れ替わりに対する手法および, イベントドリブン法によるパケットの受信間隔での送受信を行う手法を適用し，実際の小形 DC モータ速度制御システムを用いて実験を行った.

実験を行った結果，タイムスタンプによって最新のデータを用いて制御することでステップ応答のオーバーシュートが小さくなる傾向が見られた．また，イベントドリブン法による受信間隔での送受信はステップ応答にあまり改善が見られなかった.

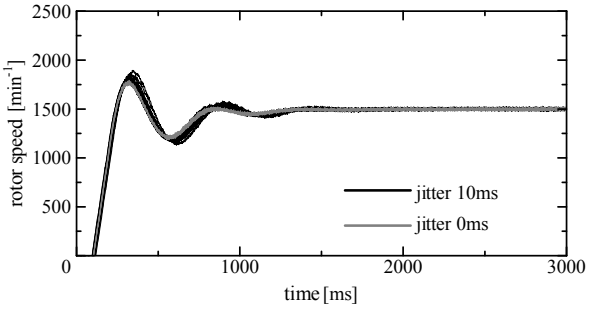
一方，それらを組み合わせた送受信法で実験を行ったところ，揺らぎによるばらつきを減少が減少すると同時に，全体の傾向としてオーバーシュートが小さくなる傾向があることがわかった．今後は，この制御法を用いて制御の改善を行うことが可能か検討していく.



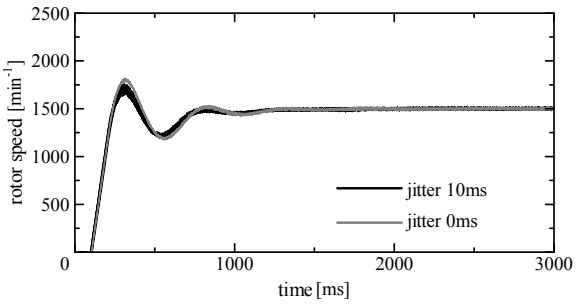
(a) Time-driven＋Non-timestamp



(c) Event-driven＋Non-timestamp

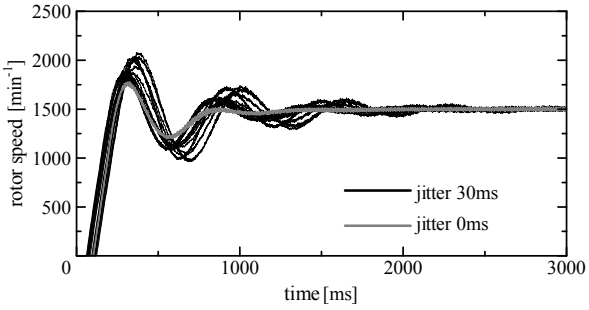


(b) Time-driven＋Timestamp

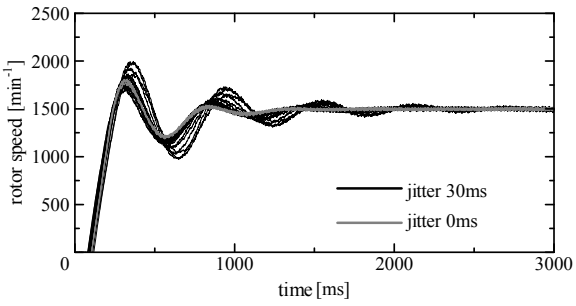


(d) Event-driven＋Timestamp

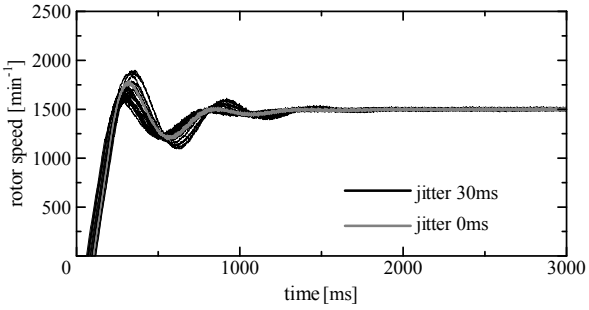
(A) Time delay 50ms, Jitter 10ms



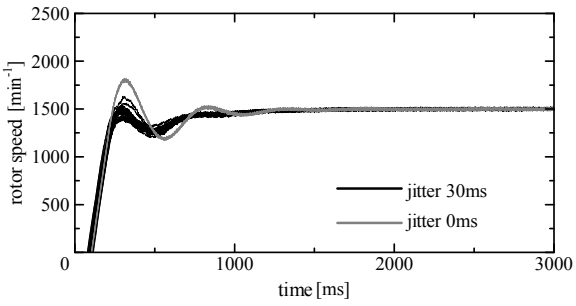
(a) Time-driven＋Non-timestamp



(c) Event-driven＋Non-timestamp



(b) Time-driven＋Timestamp



(d) Event-driven＋Timestam

(B) Time delay 50ms, Jitter 30ms

図 5 送受信法(a)-(d)を用いた速度制御によって得られたステップ応答の比較

Fig.5 Comparison of step responses obtained by speed control using the transmitting and receiving methods (a)-(d).

参考文献

1) Y. Tipsuwan and M.-Y. Chow: Gain Scheduler Middleware: A Methodology to Enable Existing Controllers for Networked Control and Teleoperation-Part I: Networked Control, IEEE Transactions on industrial electronics, **51**-6, 1218/1227 (2004)

2) Y.-B. Zhao, G.-P. Liu and D. Rees: Actively Compensating for Data Packet Disorder in Networked Control Systems, IEEE Transactions on circuits and systems, **57**-11, 913/917 (2010)

3) W. Zhang, M. S. Branicky, and S. M. Phillips: Stability of Networked Control Systems, IEEE Control Systems Magazine, **21**-1, 84/99 (2001)

4) K. Matsuo, T. Miura and T. Taniguchi: Speed Control of a DC Motor System through Delay Time Variant Network, SICE-ICASE International Joint Conference 2006, TA15-3, 399/406 (2006)

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