IPネットワークを介した小形DCモータのパケット損失を考慮した一速度制御法

工学資源学部電気電子工学科　**松　尾　健　史**

1. はじめに

近年，インターネット，すなわち，IP（Internet Protocol）ネットワークの普及は急速であり，世界のどこからでもアクセスできる環境は日々整備されている。他の通信手段に比べ通信や設備コストが低く抑えられるため，例えばIP電話など各種応用例が増えている。また，IPネットワークを介して，離れた所にあるセンサの情報を取得することや，離れた所にある機器に命令の信号を送ることなど，遠隔で操作することが可能であり，これらに関する応用も期待できる。この一例として，IPネットワークを介したモータの制御法開発があり，近年この制御法開発に関する報告が増えてきている（例えば[1]など）。

しかし，IPネットワークの通信は，パケットと呼ばれるひとまとまりにして送受信するパケット交換方式である。このため，１本の回線を共有して利用することができる利点はあるが，ネットワークの負荷が大きくなると，通信遅延時間が大きくなり，さらにその時間が変動する欠点がある。また，パケット損失が生じることもあり，これらが制御システムに悪影響を与える要因となる。

従来の制御システムにおいて，通信遅延時間はむだ時間要素として取り扱うことが考えられるが，これは一定の遅延時間としてしか扱えない。IPネットワークのように通信遅延時間が変動する場合，単にむだ時間として取り扱うだけでは不十分である。そのため，制御システム上でこの時間をどのように取り扱うか重要になる。これに対して，変動する通信遅延時間を統計的に取り扱う考え方があり，例えば，指数分布的に振る舞うと仮定して変動するその時間を捕らえ，PI制御器のゲインを調整する制御スキーム[1]などが提案されている。他には，制御信号を受信側のバッファに一旦蓄えて，一定の通信遅延時間にするジッタバッファの手法[2]も考えられる。この手法を用いると一定時間として扱えるため，むだ時間として扱うことができ，制御システムの設計が容易になる。さらに，もう一つ考慮すべき事項はパケット損失である。もしネットワーク上でパケット損失が生じれば，本来サンプリング時間ごとに得られるはずの制御信号が得られないため，制御性能劣化の要因となりえる。このため，これを考慮した制御法の開発も重要である。

本稿では，筆者のグループが研究を行ったIPネットワークを介した小形DCモータのパケット損失を考慮した一速度制御法[3][4]の紹介をする。具体的な制御スキームは，変動する通信遅延時間の影響を低減するため，文献[2]のジッタバッファの手法を用いて一定の通信遅延時間にする。次に，パケット損失の影響を受けなくするため，各サンプリング時に送信する情報に，

前のサンプリング時の情報を付加する手法を導入する。このネットワークを介した送受信を以上のように処理した上で，むだ時間により遅れる出力を予測して制御をするため，スミス補償型PI制御を用いる。最後に，パケット損失に関する本手法が本制御システムで有効であること，また，本制御スキームがこのシステムでは有用であることを実験により確認する。

2. 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

本稿で考えるIPネットワークを介した制御システムのブロック線図を図１に示す。図で示されるように，制御器と制御対象の間が，IPネットワークを介して接続されているシステムとして考える。ここで，$R(s)$ は目標値，$Y(s)$ は出力，$E(s)$ は目標値と出力との偏差，$U(s)$ は操作量を示す。また，一般的に通信遅延時間は，制御システムにおいてむだ時間として取り扱うことが考えられる。ここでは便宜的に図１のように$e^{-\alpha}$，$e^{-\beta}$として表現することにする。しかし，IP ネットワーク上では通信遅延時間は変動するため，一定時間が前提であるむだ時間として表現することはできない。また，パケット損失が生じる場合はこの制御システム上で表現することが難しい。

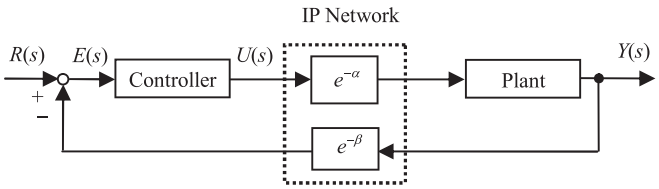

図1　IPネットワークを介した制御システムのブロック構図

前述のIP ネットワークを介した制御システムを元に，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を構築する。図２にその実験装置を示す。図２においてComputer１は図１における制御器の役割を果たす。また，DCモータは図１のおける制御対象である。Computer２はDA変換器，AD変換器，および，Computer１とIP ネットワーク介して通信するだけの機能しか持たないものとする。制御の流れは以下の通りである。Computer１によって制御器により計算された電圧指令値がIP ネットワークを通してComputer２へ送信される。Computer２はその値をDA変換器，増幅器を介してDCモータに印加することによりモータが駆動する。その回転速度をタコジェネレータにより検出して，Computer２はその値を低域フィルタ，AD変換器を介して取り込み，それをComputer１へ送信する。IP ネットワークを介してComputer１が受信し制御器上で次の指令値が計算される。

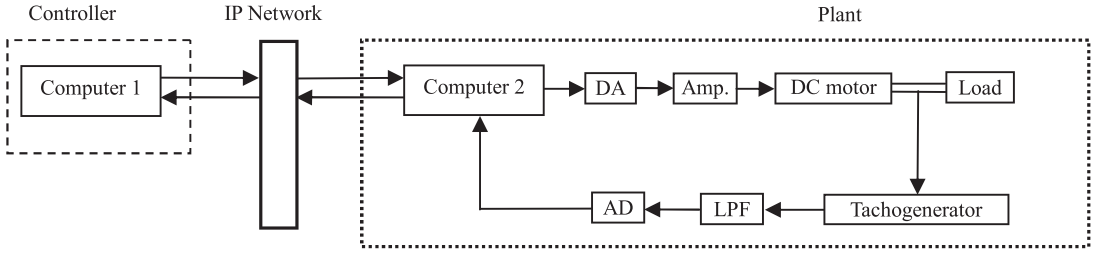

図2　IPネットワークを介した小形DCモータ速度制御システムの実験装置

表1　小形DCモータの仕様

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

用いる小形DCモータは山洋電気R301T-011であり，その仕様を表1に示す。タコジェネレータの仕様は3/1000 V/min^{-1}である。また，このモータには慣性モーメント3.5×10^{-5}kg・m^2の慣性負荷が装着されている。制御器はComputer 1上でソフトウェアによって実装する。具体的な制御器の設計については次の章で述べることにする。Computer 1とComputer 2上では，それぞれサンプリング時間1 msで処理される。また，IPネットワークを介してComputer 1とComputer 2間で制御信号が送受信される。サンプリング時間毎に，Computer 1からComputer 2へはモータに印加する電圧指令値の信号を，Computer 2からComputer 1へはタコジェネレータにより検出された回転速度の信号を送信する。IPネットワークで用いられる代表的なプロトコルとして，TCP（Transmission Control Protocol）とUDP（User Datagram Protocol）が挙げられるが，このシステムでは即応性に優れたUDPを用いることにする。ここで，TCPとUDPの大きな違いは，TCPでは受信できたかどうか確認を行い，もし受信できていなければ再送信するプロトコルであるが，UDPではそのような受信確認を行わないプロトコルである。そのため，ネットワーク上で何らかの原因でパケットが損失した場合，受信側ではその部分のパケットが損失したままになる。その代わり，TCPに比べ処理が早く行うことができ，リアルタイム性が求められるときによく用いられる。

3. 変動する通信遅延時間とパケット損失を考慮した制御法

3-1 制御スキーム

図3に示されるように，本スキームはスミス補償型PI制御を用いて行う。

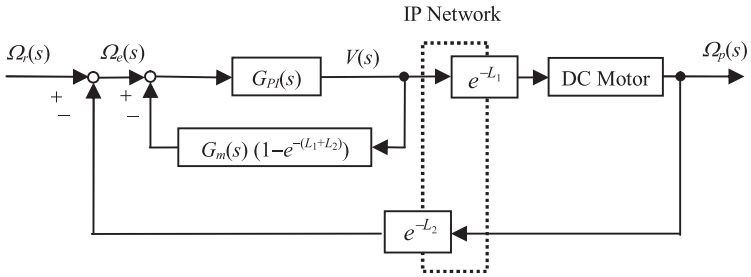

図3　スミス補償型PI制御を用いたIPネットワークを介した
小形DCモータ速度制御システムのブロック線図

ここで，むだ時間以外の表記の意味を以下に示す。

$\Omega_r(s)$：目標回転速度 [min^{-1}]

$\Omega_p(s)$：回転速度 [min^{-1}]

$\Omega_e(s)$：目標回転速度と回転速度の偏差（$=\Omega_r(s)-\Omega_p(s)$）[min^{-1}]

$V(s)$ ：印加電圧値 [V]

$G_{PI}(s)$：PI制御器

$G_m(s)$：DCモータのモデル

ここで，$G_{PI}(s)$ は伝達関数

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

で表され，K_p は比例ゲイン，K_I は積分ゲインを表す。

変動する通信遅延時間やパケット損失の影響を受けないようにするため，IPネットワーク上の送受信時に 2 つの手法を用いる。具体的には，文献[2]のジッタバッファの手法を用いて，変動する通信遅延時間を一定にする操作をする。また，パケット損失もシステムへ影響を及ぼすが，詳細は次の3-2節で説明することにする。これらの手法を用いることで，制御信号が損失しない一定時間のむだ時間を含むシステムとして取り扱うことができる。そこで，図 3 の制御システムのように，スミス補償型PI制御を用いて，むだ時間により遅れて出る出力の予測制御が可能になる。

IPネットワークの部分は，前述の手法を用いることで一定のむだ時間で送受信することが可能になり，ここでは制御器からモータへのむだ時間を L_1 [s]，モータからコントローラのむだ時間を L_2 [s]とし，図 3 で示されるように，それぞれむだ時間要素 e^{-L_1}，e^{-L_2} として表現する。このとき，スミス補償器は L_1+L_2 のむだ時間として近似し設計する。また，スミス補償器を用いるときはモデルが必要になるが，用いたDCモータのモデルは次の式で表現することができるため，これを $G_m(s)$ とする。

$$G_m(s)=\frac{8.0\times10^6}{s^2+8.8\times10^3 s+3.9\times10^4}.$$

3-2 パケット損失の対策法

前節で述べたようにIPネットワークを介した制御を行う場合，変動する通信遅延時間の他にパケット損失による制御システムへの悪影響を考慮する必要がある。パケット損失の対策としては，図 4 に示されるように，サンプリング時間毎に，現在のサンプリング時間時のデータの他に，前のサンプリング時に送信したデータも一緒に付加して送信する手法を用いる。例えば図 4 のように，Computer 1 が 101番目のサンプリング時に送信した信号が欠損しても，次の102番目のサンプリング時に 101番目の信号を修復することができる。変動する通信遅延時間のため制御信号が受信側に到着する間隔が異なる場合，受信側で前節述べたジッタバッファの手法を用いることにより一定のむだ時間にするが，ジッタバッファに蓄えられている間にパケット損

失により失われたデータを回復できれば，制御システムとしてはパケット損失による影響を受けない。そのため，このシステムでは遅延時間だけを考慮して設計すればよい。

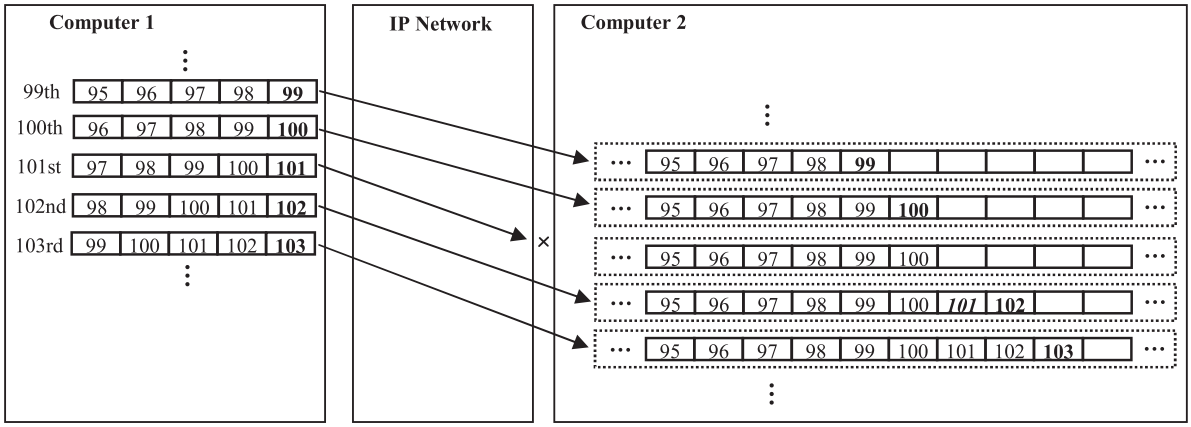

図4　パケット損失を修復する手法

4. 速度制御実験とその結果

本手法が有用であることを確かめるため，速度制御実験を行う。その前に本実験で用いるIPネットワークの構成について説明する。

4-1 IPネットワークの構成

図 2 で示した実験システムにおいて，そのIPネットワーク部分の構成を図 5 に示す。図より，Computer 1 と Computer 2 の間には，ネットワークエミュレータである Nist Net [5][6] がインストールされたコンピュータをルータとて使用する。この Nist Netは設定パラメータを変えることで，通信遅延時間，その揺らぎ，パケット損失を疑似的に作り出すことができる。実際のネットワークより，それをエミュレートした疑似ネットワークで実験を行った方が性能の評価を行い易いため，これを用いて以下実験を行うことにする。

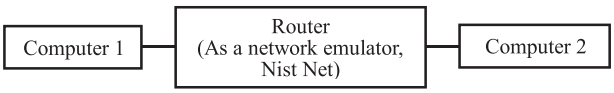

図5　本実験で用いるIPネットワークの構成

4-2 実験1 パケット損失への有効性を検証する実験

どちらも図 3 で示した実験システムで速度制御実験を行う。ここでは，パケット損失を修復する 3-2 節の手法が有効か検証するために，次の 2 つ場合で実験を行う。これらの実験の違いは， 3 章で説明したIP ネットワーク部の制御信号の送受信に用いた 2 つの手法のうち，3-2 節で説明したパケット損失による信号の欠損を回復する手法を適用するかしないかである。

（実験1-1） 3章の制御スキーム（スミス補償型PI制御を用いている）を用いた制御システムで実験

（実験1-2） 3章の制御スキームのうち3-2節の手法を用いない制御システムで実験

ここでは，どちらの実験も目標回転速度1000 min^{-1}で行い，PI制御器のゲインはどちらもK_P =0.020とK_I =0.110とする。IPネットワークにおいて，ルータを通過するパケットの5%が損失する設定をNist Netで行い，その環境下で実験を行う。実験1のネットワーク構成では，Computer 1とComputer 2間の往復遅延時間（RTT: Round Trip Time）は1ms以下である。これはサンプリング時間より短く，むだ時間がないとみなせる。ただし，ジッタバッファを設けるため，ここでは試行錯誤的に $L_1 = L_2$ =0.010とする。また，実験1-1において，一度に送信する信号は10サンプリング時間とする。

図6に速度制御実験で得られるステップ応答波形を示す。実験結果より，制御信号が損失したため，実験1-2のステップ応答波形が乱れていることが分かる。しかし，実験1-1においては，そのような現象は起きていない。これは，あるサンプリング時間時に，過去のサンプリング時間時に得られたデータも常に一緒に送るため，3-2節で説明したように，IPネットワーク部のルータを通過するとき一定割合でパケットが損失しても，後のデータにより欠損データが修復されているからである。以上実験1-1より，3-2節の手法はパケット損失が起こるネットワークにおいて有効であることが分かる。

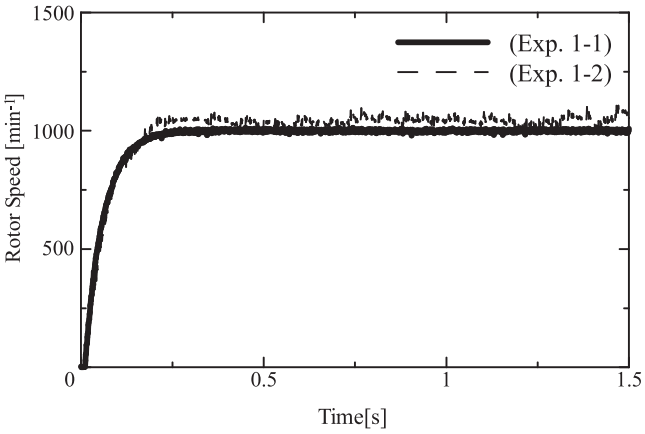

図6 実験1-1と1-2のステップ応答波形

4-3 実験2 スミス補償器を用いた本スキームの有用性を確認するための実験

本スキームにおいて，スミス補償器を用いたことの有用性を確認するために，次の実験2-1と実験2-2の実験を行う。なお，実験1と同様，本実験でも目標回転速度1000 min^{-1}で行う。

（実験2-1） 3章の制御スキーム（スミス補償型PI制御を用いている）を用いた制御システムで実験

（実験2-2） PI制御器のみを用いた制御システムで実験

ルータにインストールされているNist Netは，初期設定のままで得られるHeavy-Tail分布にお

いて，そのパラメータが平均0.050sec，標準偏差0.010sec，パケット損失3%に設定されている。このIPネットワークの下で実験を行う。ここで，実験2-2のシステムは図3のスミス補償器部分がないPI制御器のみを用いた制御システムである。

これらの条件で速度制御をそれぞれ行い，そのときのステップ応答波形を見ることにする。実験2-1において $L_1 = L_2$=0.150とし，10サンプリング時間の値を送信する。実験2-1において，PI制御器のゲインは K_p = 0.020，K_I = 0.110とする。実験2-2において，ステップ応答波形が振動的にならないようにゲインを試行錯誤的に調整し，その結果 K_p = 0.007，K_I = 0.018とした。このとき，2つの実験におけるそれぞれの応答波形の結果を図7に示す。実験結果より，明らかに立ち上がり時間は実験2-2より実験2-1が早いことが分かる。また，実験2-2ではIPネットワークの通信遅延時間が変われば，それに応じてPI制御器のゲインを調整しなければいけないが，実験2-2ではスミス補償器を用いているため，ゲインの調整を必要としない。このため，PI制御器だけを用いて制御するより，本スキームを用いて制御する方が有用であるといえる。

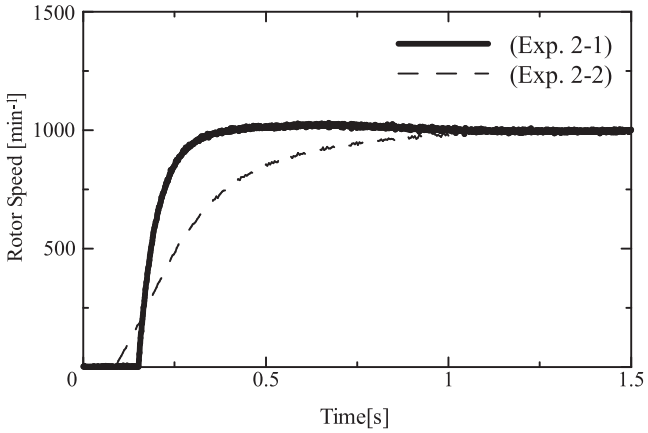

図7　実験2-1と2-2のステップ応答波形

5. おわりに

本稿では，IPネットワークを介したDCモータ速度制御システムにおいて，IPネットワーク上で変動する通信遅延時間，および，パケット損失による制御性能の劣化を抑制するための制御スキームを紹介した。まず，変動する通信遅延時間をジッタバッファの手法を用いて一定の遅延時間にした。次にパケット損失に対して，サンプリング時間ごとにパケットを送信するとき，現在のサンプリング時間の値に過去のサンプリング時間の値を付加することで，パケット損失が起きてもジッタバッファに蓄えられている間に損失時に失った値を修復できるようにした。その上で，むだ時間により遅れて出る出力を予測するため，スミス補償型PI制御を用いた。本手法の有効性および有用性について，速度制御実験を行うことで検証した。その結果，パケット損失に対して有効であること，またスミス補償器を用いることで，PI制御器のゲインを調整しなくても制御でき，有用であることも確認できた。

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*, Vol.51, No.6, pp.1218-1227 (2004)

[2] 加藤, 西, 大西: 「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, Vol. 126, No. 12, pp.1737-1738 (2006)

[3] K. Matsuo, T. Miura and T. Taniguchi: “A Speed Control Method of Small DC Motor through IP Network Considering Packet Loss”, *IEEJ Transactions on Electrical and Electronic Engineering*, Vol.2, No.6, pp.657-659 (2007)

[4] 松尾，三浦，谷口：「IPネットワークを介した小形DCモータ系の一速度制御法」，電気学会回転機リニアドライブ合同研究会，RM-08-43 / LD-08-38 (2008)

[5] Nist Net: http://www-x.antd.nist.gov/nisinet/

[6] M. Carson and D. Santay: “Nist Net - A Linux-based Network Emulation Tool”, *ACM SIGCOMM Computer Communication Review*, Vol.33. No.3, pp. 111-126 (2003)