工学実験指導書 4J21　ＤＣサーボモータの制御

2000.9.1　小坂敏文

１　目的

本実験の目的は次の２つである．第１に制御工学で学習したフィードバック制御理論をＤＣサーボモータの制御という具体例を通して学ぶ．具体的にはセンサ（ロータリーエンコーダ）の読み取り，フィードバック制御，アクチュエータ（モータ）駆動について学ぶ．第２に位置フィードバックおよび速度フィードバックの役割を調べ，当該実験装置における最適なフィードバックゲインを求める．求められたフィードバックゲインによる制御動作の再現性を確認する．

２　概要

本テーマは次のような手順で行ないます．

（１）ＤＣサーボモータ制御実験装置とテキストの図の確認
（２）マニュアル（可変抵抗）によるモータ指令値発生とモータ駆動
（３）ロータリーエンコーダによるモータ回転角の読み出し
（４）コンピュータ指令値送出によるモータ指令値発生とモータ駆動
（５）フィードバック制御

３　使用機器

本テーマは次のような機器を用います．

（１）パーソナルコンピュータＰＣ９８０１　　１台
（２）ＤＣサーボモータ制御実験装置　　１台
（３）実験用プログラム　count.c, drive0.c drive.c feedback.c　とその*.exe
　　ユーティリティプログラム　grview.exe

４　注意事項

（１）モータ回転中は回転部分に手を触れないこと．手で止めようとしてはいけない．
（２）テキスト中にスイッチを切り替える等の指示があるが，どのスイッチか図で確かめるなさい．
（３）メインスイッチは手順に従ってＯＮ－ＯＦＦのこと．

５　ＤＣサーボモータ制御実験装置

５．１　装置の説明

図１にＤＣサーボモータ制御実験装置を示す．「ＤＣサーボドライバ基板」「ＤＣサーボカウンタ基板」という説明が本プリントに現われるので，どの部分のことか確認しなさい．
また本実験で手を触れるスイッチ等は全部で次の３ヵ所である．どこにあるか確かめておきなさい．

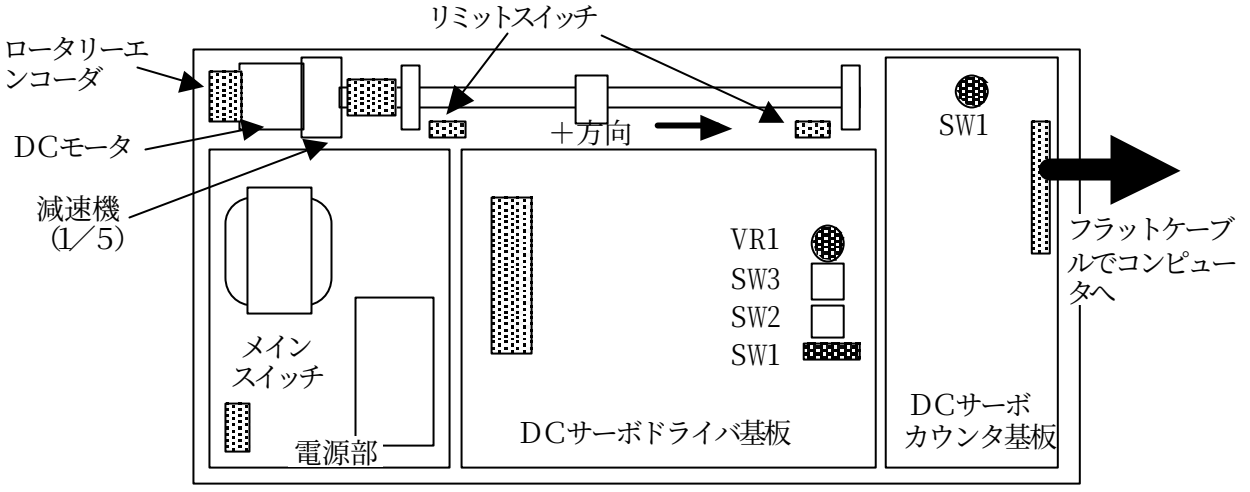


図１　ＤＣサーボモータ制御実験装置

（1）「ＤＣサーボドライバ基板」の SW１　　動作モードを切り替える「モード切替 SW」
（2）「ＤＣサーボドライバ基板」の VR１　　マニュアルモードでの速度指令値発生用の「速度指令 VR」
（3）「ＤＣサーボカウンタ基板」の SW１　　カウンタをリセットするための「カウンタリセット SW」
なお「ＤＣサーボドライバ基板」の SW２はＯＮ，SW３は PWM に固定しなさい．

５．２　電源投入
（1）コンピュータとフラットケーブルで結ばれているのを確認して，コンピュータの電源を入れる．必ず，コンピュータの電源を先に入れること．
（2）ＤＣサーボモータ制御実験装置の「ＤＣサーボドライバ基板」の SW１を２の位置にしてから，メインスイッチをＯＮにする．いきなりモータが回転しても驚かなくてよい．モータは適当な位置で止まるはずである．

６　センサとアクチュエータについての説明
６．１　ロータリーエンコーダとカウンタ
モータの回転角度センサとしてよく用いられているセンサはロータリーエンコーダである．ロータリーエンコーダは図２に示すように全周にスリットの付いた円板とスリットによって光がさえぎられたり通過したりする光センサより成り立っている．光センサは２つあり，位置がずらして取り付けてある．軸が正転している時は，センサＡセンサＢは図３正転時のように信号を出し，逆転時には図３逆転時のように信号を出す．センサＢが High の時センサＡのエッジが正か負かを観測することにより，正転逆転の判断がつき，何パルス回転したか測定することができる．実験装置のロータリーエンコーダはモータ１回転につき１００パルスを発生する性能を持っている．またモータの回転は減速機により１／５になっているため，実験装置の見えている軸１回転では５００パルスを発生する．実験装置の「ＤＣサーボカウンタ基板」には１２ビット＋符号ビットを持つカウンタが乗っている．このカウンタの１２ビットの内容は「ＤＣサーボカウンタ基板」の１２個の赤い LED に表示されている．

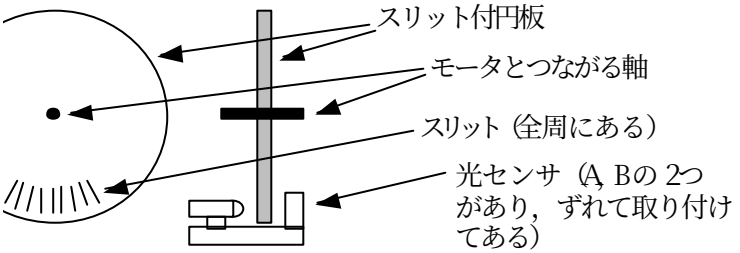


図２　エンコーダの仕組

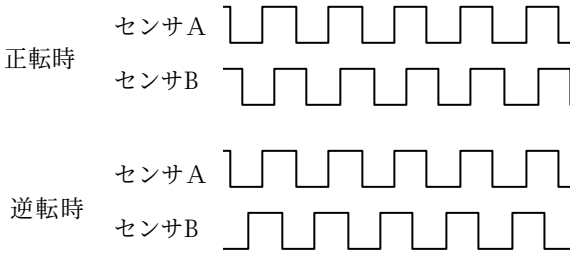


図３　エンコーダの信号

６．２　アクチュエータ
モータのようにものを動かす機構のことを一般にアクチュエータと呼ぶ．本実験装置ではＤＣサーボモータが，使われている．モータを動かす回路のことをモータドライバと呼ぶ．モータを状況に応じて正逆転させるためには原理的に図４ａのような構成にすればよい．スイッチのどちらかをＯＮにす

れば，モータを正逆転させることができる．ところで図４ａのスイッチをトランジスタで置き換えると図４ｂとなる．これもどちらかのトランジスタのベースに信号を与えればよい．図４ｂの回路を単一電源で実現するには図４ｃのようにすればよい．ここではトランジスタ１と４あるいは２と３の片方の組み合わせに信号を与えればよい．この実験装置のドライバは図４ｂの形式のものである．
トランジスタを駆動する信号形式は，リニアドライブと PWM ドライブの２通りがある．

（１）リニアドライブ
信号電圧を変化させ，トランジスタの増幅作用を利用することで，モータに与える電圧をコントロールする．この場合トランジスタは抵抗の役割を果たすので発熱による電力損失が多い．

（２）PWM ドライブ
図４ａのスイッチ１を高速にＯＮ－ＯＦＦを繰り返し，ＯＮになっている時間とＯＦＦになっている時間の比を変化させると，見かけ上モータに与える電圧を変化させたことになりモータの回転速度をコントロールすることができる．この作業を図４ａのトランジスタで高速（１kHz くらい）に行うとモータのコントロールが可能である．図５のように，トランジスタがＯＮになっている時間とＯＦＦになっている時間の比をデューティ比と呼び，デューティ比を変化させることにより，モータの回転速度をコントロールする．このドライブ方法は PWM（Pulse Width Modulation，パルス幅変調）ドライブと呼ばれる．

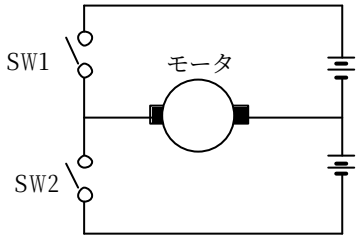


図４ａ　モータドライバモデル１

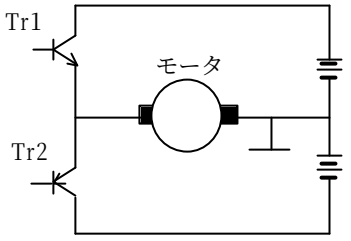


図４ｂ　モータドライバモデル２

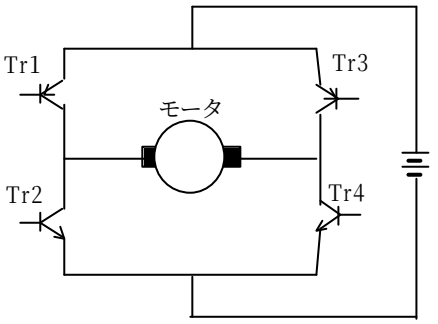


図４ｃ　モータドライバモデル３

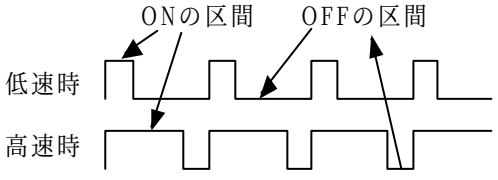


図５　PWM ドライブでトランジスタに与える信号

７　マニュアルによる速度指令とカウンタ読み出し

７．１　マニュアルによる速度指令

（１）「ＤＣサーボドライバ基板」の SW１「モード切替 SW」を１に切り替える．この時モータが急に動き出しても驚かなくてもよい．リミットスイッチで止まるようになっている．SW１を１に切り

替えるとマニュアルモードとなり，モータに与えられる電圧を手動で変更できる．
（２）「ＤＣサーボドライバ基板」の VR１（黄色のつまみ）「速度指令 VR」を適当に回してみよう．時計回りに回すと負荷はモータに近づく方へ，反時計回りに回すと負荷はモータから遠ざかる方へ移動するはずである．VR１の角度に応じた電圧がモータに加わるため，それに応じた速度で移動していることを確認しなさい．

重要　この後の８以降の実験でモータの回転により負荷の位置が悪い場合，ここで行なったように「ＤＣサーボドライバ基板」の SW１「モード切替 SW」を１に切り替え，「ＤＣサーボドライバ基板」の VR１（黄色のつまみ）「速度指令 VR」を適当に回して負荷を都合のよい位置まで動かせばよい．

７．２　マニュアルによる負荷の駆動とカウンタ読み出し

（１）プログラム counter.exe を実行する．（プログラムリスト１参照）
（２）画面表示に従い，「ＤＣサーボドライバ基板」の SW１「モード切替 SW」を１にする．
（３）画面表示に従い，「ＤＣサーボドライバ基板」の VR1「速度指令 VR」を調節して，負荷を中央で止める．
（４）画面表示に従い，「ＤＣサーボカウンタ基板」の SW1「カウンタリセット SW」を押す．これにより，「ＤＣサーボカウンタ基板」内のエンコーダ信号のカウンタがリセットされる．同時に１２この LED も消え，０を表わす LEDZ が点灯する．
（５）画面表示に従い，「ＤＣサーボドライバ基板」の VR1 を動かすと，負荷が移動し，画面上に負荷の変位（カウンタの値，何パルス分軸が回転したかがわかる）と負荷の速度（2msec の間にどのくらいカウンタの値が増加したか）が表示される．
（６）「ＤＣサーボドライバ基板」の VR1 を調節して，変位１０００のところに止めてみよ．１０００パルス分の移動なので軸は２回転する．かなり難しいはずである．どこに止められたか記録し報告書作成に備えなさい．
（７）正負の最高速度はどのくらいの値になるか記録し報告書作成に備えなさい．

プログラムリスト１　counter.c の要点

```
#define COUNTIN   0xd1 /*カウンタ入力ポート*/
#define INTERVAL   2.0 /*割り込み間隔 2.0msec*/

volatile int counter=0,velocity=0;

void main()
{
        char c=0;
        initper();/*インターフェイスボードの初期化*/
        printf("１．ＤＣサーボドライバ基板の SW1 を１にして下さい．¥n");
        printf("２．VR1 を調節して，負荷を中央で止めて下さい．¥n");
        printf("３．ＤＣサーボカウンタ基板の SW1 を押して下さい．¥n");
        printf("４．ここまで済んだらスペースキーを押して下さい．¥n");
        getch(); printf("¥n");
        printf("ＤＣサーボドライバ基板の VR1 を調節して，負荷を動かして下さい．¥n");
        printf("スペースキーを押すとプログラムは終了します．¥n");
        printf("        変位     速度¥n");
        timer98(1,INTERVAL,timeintcounter);/*タイマー割り込み許可*/
        while (c!=' '){
                printf("    %6d %6d¥r",counter,velocity);
                if (kbhit()) c=getch();
        }
        timer98(0,INTERVAL,timeintcounter);/*タイマー割り込み禁止*/
}

void interrupt far timeintcounter(void)/*割り込みルーチン*/
/*カウンタの値を読み出して counter と velocity に値を代入します*/
```

```
{
        static int lastx=0,lastcounter=0;
        int x,pn;
        disable();

        x=inportb(COUNTIN);
        pn=x&02;
        x>>=2; x=~x;
        if(pn) x=-x;
        x&=0x03f;
        if (0x10<x-lastx) counter=x+(counter&0xffc0)-0x40;
        else if (x-lastx<-0x10) counter=x+(counter&0xffc0)+0x40;
        else counter=x+(counter&0xffc0);
        velocity=counter-lastcounter;
        lastx=x;
        lastcounter=counter;

        outportb(0,0x20);/*send EOI*/
        enable();
}
```

プログラムの説明
（1）「タイマー割り込み許可」と「タイマー割り込み禁止」の区間で 2msec **ごとに「割り込みルーチン」が起動し，「ＤＣサーボカウンタ基板」内のエンコーダ信号のカウンタを読み込み，**counter **と** velocity **の変数にしまっている．メインではキーボードを監視しながら** counter **と** velocity **の変数を表示し続ける．**
（2）「割り込みルーチン」では「ＤＣサーボカウンタ基板」内のエンコーダ信号のカウンタを読み込むが，このカウンタは１２ビットの絶対値をカウントし，別に符号ビットを持っている．インタフェイス部では図６に示すようにこのカウンタの下位６ビットの 0－1 **の反転したものと符号ビットがコンピュータから見えるようになっている．「割り込みルーチン」中** x=inportb(COUNTIN);**では次の内容が読み出されている．**

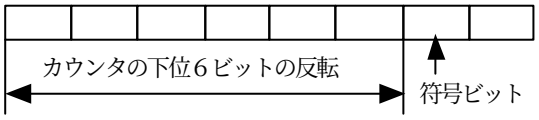


図６カウンタ読み出しインタフェイス

すなわち，もし，パルスカウントが１０ならば，カウンタは 000000001010 **符号ビット** 0 **でコンピュータからは** 1101010x **が読み出され，パルスカウントが－１５ならカウンタは** 000000001111 **符号ビット** 1 **で** 1100001x **が読み出される．**
＜このインタフェイスは設計が悪いため，ソフトウェアが苦労してカウンタの値を読み取らなければならない．通常は１６ビットあるいは２４ビットのカウンタを直接読むだけですむ設計とする＞

また速度は次の式で算出される．単位は「パルス／２ｍｓｅｃ」である．
速度＝現在のカウンタの値－2msec **前のカウンタの値**

８　コンピュータからの指令値によるモータ駆動
７では「ＤＣサーボドライバ基板」の SW**１を１切り替え，マニュアルモードとして，モータに与えられる電圧を手動で変更した．ここでは「ＤＣサーボドライバ基板」の** SW**１を３切り替え，コンピュータ指令値モードとして，モータに与えられる電圧をコンピュータからの指令値で変更する．**

８．１　drive0.exe **による駆動**
（1）「ＤＣサーボドライバ基板」の SW**１「モード切替** SW**」を３に切り替える．**

（２）drive0.exe を実行する．（プログラムリスト２参照）
（３）適当な値を入力してモータを動かせ．値の入力のたびに指令値は変更される．
（４）正負両方につき，指令値の絶対値をどこまで小さくしたら，モータは摩擦に負けて指令値通り動かなくなるか限界をさぐろう．動くことのできる最小の値を正負それぞれについて求め，記録し報告書作成に備えなさい．

プログラムリスト２　drive0.c の要点

```
#define DAOUT 0xd3

void main()
{
        int data;
        initper();/*インターフェイスボードの初期化*/
        outportb(DAOUT,0x80);/*モータ停止値の出力*/
        printf("ＤＣサーボドライバ基板のＳＷ１を３に合わせて下さい．￥n");
        printf("指令値に応じた速度で動作します．￥n");
        printf("指令値は-128～127 の範囲の値です．￥n");
        printf("範囲外の値を入力したらプログラムは終了します．￥n");
        printf("-128.....0.....127¥n");
        printf(" 左向き　　モータ　　右向き￥n");
        printf("最高速度　　停止　　最高速度￥n￥n");
        printf("指令値(-128～127,範囲外は終了)＝ ");
        scanf("%d",&data);
        data+=0x80;
        while (0<=data&&data<=0xff) {
                outportb(DAOUT,data);
                printf("指令値(-128～127,範囲外は終了)＝ ");
                scanf("%d",&data);
                data+=0x80;
        }
        outportb(DAOUT,0x80);/*モータ停止値の出力*/
}
```

追加説明　モータドライバとのインタフェイスでは，指令値をポートに出力するが，その値はは次のようになっている．

０・・・・・・・・・・・・・・１２８・・・・・・・・・・・・・・２５５
負の最大値　　モータ停止　　正の最大値

８．２　drive.exe による駆動
（１）「ＤＣサーボドライバ基板」の SW１「モード切替 SW」を３に切り替える．
（２）drive.exe を実行する．
（３）「＋」または「－」の適当なキーを押してモータを動かしなさい．
（４）正負両方につき，各指令値を与えた時，速度の値表示はどうなったか記録し報告書作成に備えなさい．

９　フィードバック制御
軸の回転角をロータリエンコーダとカウンタでコンピュータが知ることができ，モータへの指令値もコンピュータが出せることがわかった．そこで目的のカウンタの値に向け，モータに指令値を出すプログラムにより，フィードバック制御を行なう．目的地が現在の負荷の位置より先にある場合，正の指令値を出し，目的地が現在の負荷の位置より手前にある場合，負の指令値を出すことにする．

指令値は次式で与えられる．
指令値＝位置フィードバックゲイン×（目標とするカウンタの値－現在のカウンタの値）

また速度がある値を持つ場合速度に比例した減速のための項を加えると運動がスムースになる．そこで指令値を与える式を次のように修正する
指令値＝位置フィードバックゲイン×（目標とするカウンタの値－現在のカウンタの値）
　　－速度フィードバックゲイン×速度
さらに摩擦を考慮すると
指令値＝位置フィードバックゲイン×（目標とするカウンタの値－現在のカウンタの値）
　　－速度フィードバックゲイン×速度±摩擦分補正値（複号は前式の速度の符号を採用のこと）

実際のプログラムでは 2msec ごとの割り込み処理内で次の作業が行われている．

```
velocity=counter-lastcounter;/*速度＝現在のカウンタの値－2msec 前のカウンタの値*/
fcom=kx*(dest-counter)-kv*velocity;/*位置フィードバックゲイン×
                                        （目標とするカウンタの値－
現在のカウンタの値）*/
if (0<velocity) fcom+=minout;/*速度が正なら摩擦補正分値を加える*/
else if (velocity<0) fcom-=minout;/*速度が負なら摩擦補正分値を減ずる*/
else {
        if (0.<fcom) fcom+=minout+1;/*指令値が正なら摩擦補正分値を加える*/
        else if (fcom<0.) fcom-=minout+1;/*指令値が負なら摩擦補正分値を減ずる*/
}
if (fcom<-128.) com=-0x80;/*負の制限値*/
else if (127.<fcom) com=0x7f;/*正の制限値*/
else com=(int)fcom;
com+=0x80;/*インタフェイスに値を合わせる*/
outportb(DAOUT,com);/*ポート出力*/
```

９．１　フィードバック制御実行手順

１０００ポイント先を目標値としてステップ応答を行います．
（０）負荷が右端にいる場合は，「ＤＣサーボドライバ基板」の SW１「モード切替 SW」を１に切り替え，「ＤＣサーボドライバ基板」の VR１（黄色のつまみ）「速度指令 VR」を適当に回して負荷を左側に動かしなさい．
（１）「ＤＣサーボドライバ基板」の SW１を３に切り替える．
（２）feedback.exe を実行する．
（３）画面に従って位置フィードバックゲインを入力（０．５くらいがよいでしょう．）．
（４）画面に従って速度フィードバックゲインを入力（１０くらいがよいでしょう．）．
（５）画面に従って摩擦分補正値を入力（８．１（４）より６くらいがよいでしょう．）．
（６）ここで制御が実行されます．終わるのを待ちます．
（７）画面に従ってデータ保存用ファイル名を入力．このファイルには動作状況がテキストで保存されます．
（８）（７）で保存したファイルを DOS の type コマンドで表示してみなさい．あるいは VZ エディタで開いてみなさい．
（９）（７）で保存したファイルは grview.exe で変位波形をみることができ，プリンタへの印刷もできる．grview.exe は次のように用いる．
>grview.exe 保存したファイル名

９．２　フィードバック制御のゲインの変更

feedback.exe を用いたフィードバック制御では位置フィードバックゲインと速度フィードバックゲインを変更することにより，制御特性を変化させることができます．
次のゲインの組み合わせで実行を繰り返して，収束の様子を比べてみなさい．
位置フィードバックゲイン　0.1，　0.3，　1.0，　3.0

速度フィードバックゲイン　0.0,　1.0,　10.0
ゲイン設定と収束の様子を考慮し，その他のゲインの値の組み合わせも試して，速い収束をもつステップ応答を実現しなさい．この過程を記録し，グラフ化されたデータを印刷し，報告書作成に備えなさい．またゲインを変化させずに同じ実験を数回行なうと現象の再現性について考察できるはずです．これも行なってみなさい．この結果も記録し，グラフ化されたデータを印刷し，報告書作成に備えなさい．実行結果のファイルには各時刻の変位の他に指令置（－１２８～１２７）も記録されている．きれいに位置決めされた場合と大きな振動が残った場合について指令値の特徴について調べておくこと．（最低でも２０数枚のグラフが得られたはずです．必要ならファイルを持ち帰って別のグラフ化ツールでグラフ化してみなさい．）

２つのフィードバックゲインを変化させて得られた実際の実行例を次の図７～９に示す．

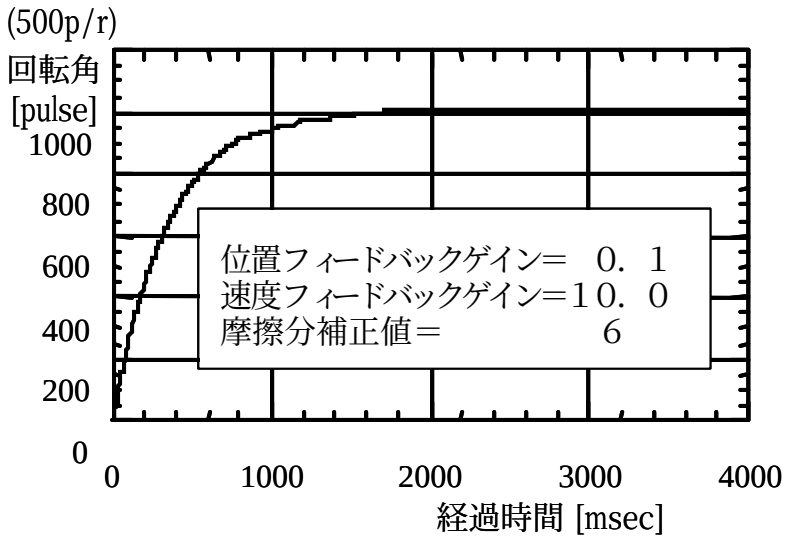


図７　ゆっくりした収束をもつステップ応答の例

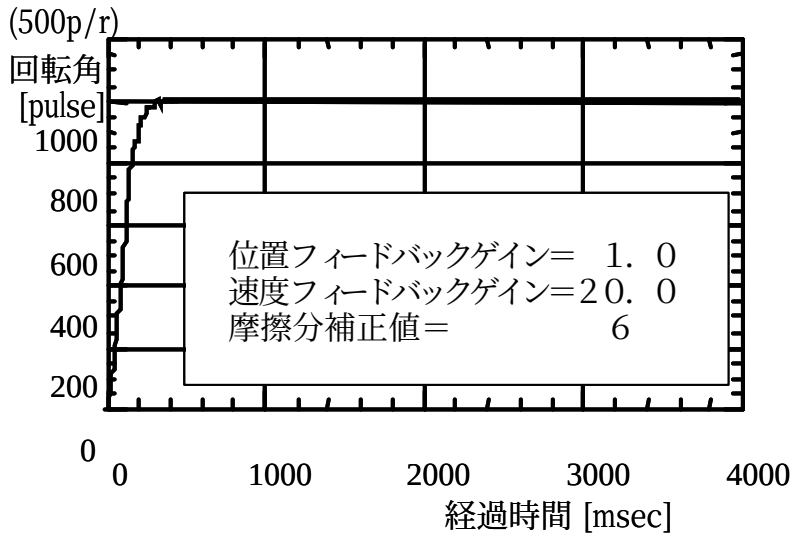
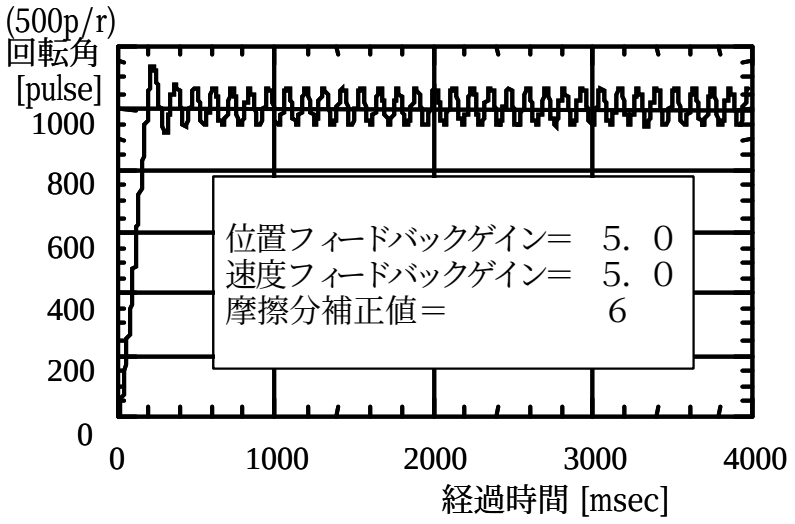


図８　速い収束をもつステップ応答の例



制御系の非線形性のため，リミットサイクル発振を起こしているのがわかる．
図９　速い収束をもつステップ応答の例

１０　実験の終了
電源オフの手順は投入の逆順で行いなさい．必ず，ＤＣサーボモータ制御実験装置のメインスイッチをオフにしてから，コンピュータの電源を切ること．

１１　終わりに
１１．１報告書の記述において次の点に留意しなさい．
（１）採点者は，この指導書の存在を知らないこととします．
（２）項目には必ず番号をつけなさい．（例えば「１．目的」「２．使用機器」...）
（３）「１．目的」から始まり，「N．結論」で終わりなさい．結論では目的に掲げた目標がどこまで達成できたかを明らかにしなさい．実験結果を記述するだけではありません．
（４）各項目には必ず本文があり，ストーリが一貫するようにしなさい．ストーリの飛躍がないように注意しなさい．
（５）必要なら図や表を付けて，本文の論旨の補強に役立てなさい．当然ながら，それらの図表は本文中から参照されることになります．
（６）図，表を用いたなら必ず図番＋図のタイトル，表番＋表のタイトルを付けなさい．
（７）図のタイトルは図の下に，表のタイトルは表の上に書きます．
（８）図は図だけで独立しも意味がわからなければなりません．そのために説明を加える場合は，上から下に向けて「図の本体」「図の説明」「図のタイトル」の順になります．（本指導プリントの図９参照）
（９）表は表だけで独立しも意味がわからなければなりません．そのために説明を加える場合は，上から下に向けて「表のタイトル」「表の説明」「表の本体」の順になります．

１１．２　報告書の考察に盛り込むべき内容
（１）横軸に位置フィードバック縦軸に速度フィードバックをとる２次元空間で，当該制御装置のふるまいの特徴を特徴ごとに領域分割して整理し，考察中に盛り込みなさい．
（２）きれいに位置決めするには，指令値がどのようになっていなければならないか考察し，考察中に盛り込みなさい．
（３）モータの回転角をどのようにしてコンピュータは読み取るのか，その原理についてまとめ，考察中に盛り込みなさい．
（４）モータドライバ回路はコンピュータによってどのように指令を受け，どのようにモータ駆動電流を流すのか，その原理についてまとめ，考察中に盛り込みなさい．
（５）ＤＣサーボモータによる位置決め制御（目標値を定めたステップ応答）はどのような原理で行われているかまとめ，考察中に盛り込みなさい．

（６）その他に気付いたことについてまとめ，考察中に盛り込みなさい．

１１．３　初歩的注意

（１）報告書中に「～しなさい」は書かないこと「～した．」にすること．
（２）最低でも２０個程度の結果（グラフ）がレポートには添付されるはずです．
（３）Ａ４用紙以外は使用しないこと．必要なら，Ａ４用紙に糊付けのこと．
（４）そのままＡ４用紙でコピー出来るようにしておくこと．