ディジタル信号処理 ／ 課題２

氏名　本村健太（もとむらけんた）
学籍番号　055762F
受講日　2007/05/11
提出日　2007/05/18

目 次

2.1 問題 1.9

◆教科書の図 1.18 の回路の動作を差分方程式で表すと以下のようになる。

(a) $y(nT) = 2x(nT) + x(nT - T) + 0.5x(nT - 2T) + 0.1x(nT - 3T)$

(b) $y(nT) = 2x(nT) + x(nT - T) + 0.5x(nT - 2T) + 0.1x(nT - 3T)$

(c) $y(nT) = a_0x(nT) + a_1x(nT - T) + a_2x(nT - 2T) + b_1y(nT - T) + b_2y(nT - 2T)$

(d) $x_1(nT) = x(nT) + b_1x_1(nT - T) + b_2x_1(nT - 2T)$
$y(nT) = a_0x_1(nT) + a_1x_1(nT - T) + a_2x_1(nT - 2T)$

C 言語を使って上記の差分方程式を計算するプログラムを書いて出力した。

◆ `p1.9.c`

```
// 2007.05.16(Wed)  18:35  完成
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>

#define MAX 100
#define R   10

void output(double *dd) {
    int i;

    printf("n     ");
    for(i=0; i<R; i++) printf("%5d    ",i);
    puts("");

    printf("y(nT) ");
    for(i=0; i<R; i++) printf("%+1.4f  ",dd[i]);
    puts("");

    return;
}


int main() {
    int t;  // t = nT
    double x[MAX];
    double x1[MAX];
    double y_a[MAX];
    double y_c[MAX];
    double y_d[MAX];
    int i;
    double a0 = 0.0,
           a1 = 0.3236,
           a2 = -0.36,
           b1 = 1.2944,
           b2 = -0.64;

    // 入力信号
    memset(x,0,MAX*sizeof(double));
    x[0] = 1;
    x[2] = -1;

    //********************************************** (a)(b)
    for(t=0; t<R; t++) {
        y_a[t] = 2.0*x[t];
        if(t>=1) y_a[t] += x[t-1];
        if(t>=2) y_a[t] += 0.5*x[t-2];
        if(t>=3) y_a[t] += 0.1*x[t-3];
    }

    puts("----------------------- (a)(b) ----------------------");
    output(y_a);

    //********************************************** (c)
    for(t=0; t<R; t++) {
        y_c[t] = a0*x[t];
        if(t>=1) y_c[t] += a1*x[t-1] + b1*y_c[t-1];
        if(t>=2) y_c[t] += a2*x[t-2] + b2*y_c[t-2];
    }
    puts("------------------------ (c) ------------------------");
    output(y_c);

    //********************************************** (d)
    for(t=0; t<R; t++) {
        x1[t] = x[t];
        if(t>=1) x1[t] += b1*x1[t-1];
        if(t>=2) x1[t] += b2*x1[t-2];
        y_d[t] = a0*x1[t];
        if(t>=1) y_d[t] += a1*x1[t-1];
        if(t>=2) y_d[t] += a2*x1[t-2];
    }
    puts("------------------------ (d) ------------------------");
    output(y_d);

    return 0;
}
```

◆実行結果

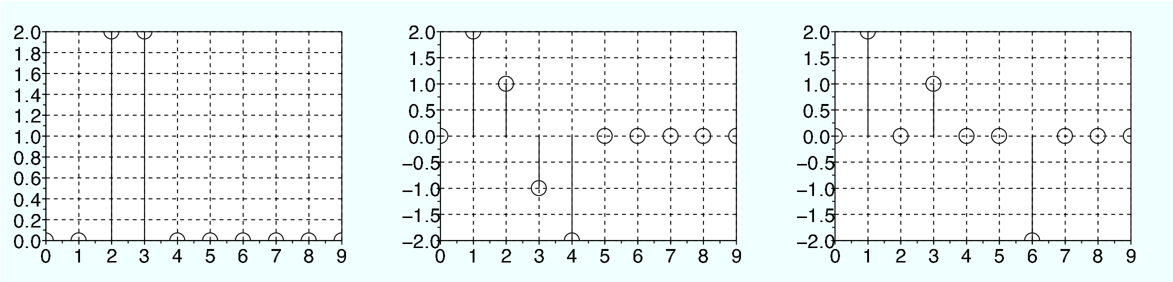
```
------------------------ (a)(b) -----------------------
n         0        1        2        3        4        5        6        7        8        9
y(nT) +2.0000  +1.0000  -1.5000  -0.9000  -0.5000  -0.1000  +0.0000  +0.0000  +0.0000  +0.0000
------------------------- (c) -------------------------
n         0        1        2        3        4        5        6        7        8        9
y(nT) +0.0000  +0.3236  +0.0589  -0.4545  -0.2660  -0.0534  +0.1011  +0.1650  +0.1489  +0.0871
------------------------- (d) -------------------------
n         0        1        2        3        4        5        6        7        8        9
y(nT) +0.0000  +0.3236  +0.0589  -0.4545  -0.2660  -0.0534  +0.1011  +0.1650  +0.1489  +0.0871
```

2.2 問題 1.11

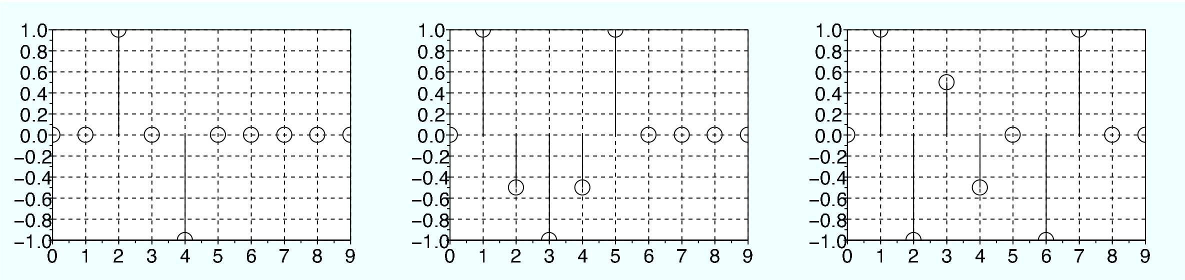
課題は 1.11(c) だけでしたが、間違えて全部やってしまいました。せっかくなので掲載します。

2.2.1 (a) $y(nT) = 2x(nT - T)$

グラフは、左から図 1.17 の信号 (a)(b)(c) を入力した場合の出力。

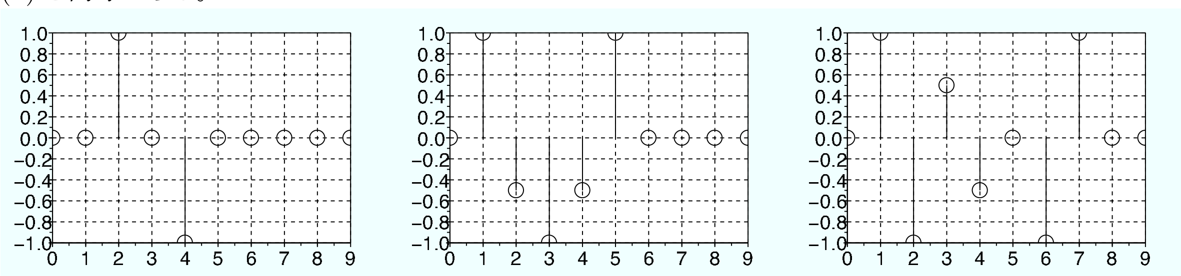


2.2.2 (b) $y(nT) = x(nT - T) - x(nT - 2T)$



2.2.3 (c) $y(nT) = x_1(nT - T)$, $x_1(nT) = x(nT) - x(nT - T)$

(b) と同等である。



2.2.4 プロットに用いた Scilab スクリプト

```
//--------------------------- 基本設定
// ***** 入力信号
xa=[0 1 1 0 0 0 0 0 0 0];
xb=[1 0.5 -0.5 -1 0 0 0 0 0 0];
xc=[1 0 0.5 0 0 -1 0 0 0 0];
// ***** 軸
xlin=linspace(0,9,10);
// ***** x0: x(nT)   ,   x1: x(nT-T)   ,   x2: x(nT-2T)   ,   x3: 一時保持
x0=0; x1=0; x2=0; x3=0;

//======================================================================== 回路 a
//--------------------------- 図 1.20(a) の回路に図 1.17(a) の信号を入力
x1=0;
for t=1:10
    y(t) = 2*x1;
    x1 = xa(t);
end;
subplot(3,3,1); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);

//--------------------------- 図 1.20(a) の回路に図 1.17(b) の信号を入力
x1=0;
for t=1:10
    y(t) = 2*x1;
    x1 = xb(t);
end;
subplot(3,3,2); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);

//--------------------------- 図 1.20(a) の回路に図 1.17(c) の信号を入力
x1=0;
for t=1:10
    y(t) = 2*x1;
    x1 = xc(t);
end;
subplot(3,3,3); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);

//======================================================================== 回路 b
//--------------------------- 図 1.20(b) の回路に図 1.17(a) の信号を入力
x1=0; x2=0; x3=0;
for t=1:10
    y(t) = x1 + x2;
    x1 = xa(t);
    x2 = x3;
    x3 = -1*xa(t);
end;
subplot(3,3,4); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);

//--------------------------- 図 1.20(b) の回路に図 1.17(b) の信号を入力
x1=0; x2=0; x3=0;
for t=1:10
    y(t) = x1 + x2;
    x1 = xb(t);
    x2 = x3;
    x3 = -1*xb(t);
end;
subplot(3,3,5); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);

//--------------------------- 図 1.20(b) の回路に図 1.17(c) の信号を入力
x1=0; x2=0; x3=0;
for t=1:10
    y(t) = x1 + x2;
    x1 = xc(t);
    x2 = x3;
    x3 = -1*xc(t);
end;
subplot(3,3,6); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);

//======================================================================== 回路 c
//--------------------------- 図 1.20(c) の回路に図 1.17(a) の信号を入力
x1=0; x3=0;
for t=1:10
    y(t) = x3;
    x3 = xa(t) + x1;
    x1 = -1 * xa(t);
end;
subplot(3,3,7); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);

//--------------------------- 図 1.20(c) の回路に図 1.17(b) の信号を入力
x1=0; x3=0;
for t=1:10
    y(t) = x3;
    x3 = xb(t) + x1;
    x1 = -1 * xb(t);
end;
subplot(3,3,8); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);

//--------------------------- 図 1.20(c) の回路に図 1.17(b) の信号を入力
x1=0; x3=0;
for t=1:10
    y(t) = x3;
    x3 = xc(t) + x1;
    x1 = -1 * xc(t);
end;
subplot(3,3,9); plot2d3(xlin,y(1:10)); plot2d(xlin,y(1:10),style=[-9]); xgrid(1);
```

2.3　問題 1.12

(a) の回路の出力を y_a とすると、差分方程式は次式となる。

$$y_a(nT) = \sum_{k=0}^{4} x(nT - kT)$$

(b) において、最初の加算器の出力を $c(n)$ とすると、

$$c(n) = x(nT) - x(nT - 5)$$

である。ここで、出力を y_b とすると、

$$y_b(nT) = c(nT) + y_b(nT - T)$$

であり、これは以下のように書くことができる。

$$y_b(nT) = c(nT) + c(nT - T) + y_b(nT - 2T)$$

これを可算和 (Σ) を使って一般化すると、

$$y_b(nT) = \sum_{k=0}^{N-1} c(nT - T) + y_b(nT - NT)$$

である。$(N = n + 1)$

$n < 0$ において $x(nT) = 0$ とすると、$y(nT) = 0$ である。よって、k に関係なく

$$y_b(nT - NT) = 0$$

であり、このため

$$y_b(nT) = \sum_{k=0}^{n} c(nT - kT)$$

となる。これに $c(n)$ を代入すると、

$$y_b(nT) = \sum_{k=0}^{n} [x(nT - kT) - x(nT - (k+5)T)]$$

である。

これを実際に展開すると $x(nT - 5T)$ 以降は消えるため、

$$y_b(nT) = \sum_{k=0}^{4} x(nT - kT)$$

となる。これは、$y_a(nT)$ に等しい。

(Q.E.D)

2.4 問題 2.1

2.4.1 問題 2.1 (2)

◆線形性の検証 (任意の実定数入力 a,b を想定)

$$\begin{aligned}
y(nT) &= x^2(nT) + x(nT+T) \\
R[ax_1(nT) + bx_2(nT)] &= [ax_1(nT) + bx_2(nT)]^2 + [ax_1(nT+T) + bx_2(nT+T)] \\
&= a^2x_1^2(nT) + 2abx_1(nT)x_2(nT) + b^2x_2^2(nT) + ax_1(nT+T) + bx_2(nT+T) \\
aR[x_1(nT)] + bR[x_2(nT)] &= a[x_1^2(nT) + x_1(nT+T)] + b[x_2^2(nT) + x_2(nT+T)] \\
&= ax_1^2(nT) + ax_1(nT+T) + bx_2^2(nT) + bx_2(nT+T) \\
\therefore R[ax_1(nT) + bx_2(nT)] &\neq aR[x_1(nT)] + bR[x_2(nT)]
\end{aligned}$$

以上より、本システムは線形システムではない。

◆時不変性の検証 (任意の時間遅れ k を想定)

$$\begin{aligned}
y(nT) &= x^2(nT) + x(nT+T) \\
y(nT-kT) &= x^2(nT-kT) + x(nT-kT+T) \\
R[x(nT-kT)] &= x^2(nT-kT) + x(nT-kT+T) \\
\therefore y(nT-kT) &= R[x(nT-kT)]
\end{aligned}$$

以上より、本システムは時不変システムである。

◆因果性の判定 (※このシステムは線形時不変システムではないので、因果性の定義に従い、「過去のサンプルだけを使って処理を行うか」をチェックする。)

$$\begin{aligned}
y(nT) &= x^2(nT) + x(nT+T) \\
n = -1 \text{ とすると、} & \\
y(-T) &= x^2(-T) + x(-T+T) \\
y(-T) &= x^2(-T) + x(0)
\end{aligned}$$

以上より、時刻 $-T$ に対して未来の信号である $x(0)$ を必要とするので、定義を満たさない。よって、このシステムは因果性システムではない。

2.4.2　問題 2.1 (4)

◆線形性の検証 (任意の実定数入力 A,B を想定)

$$\begin{aligned} y(nT) &= a^n x(nT-T) \quad (|a|<1) \\ R[Ax_1(nT)+Bx_2(nT)] &= R[Aa^n x_1(nT-T)+Ba^n x_2(nT-T)] \\ &= Aa^n x_1(nT-T)+Ba^n x_2(nT-T) \\ AR[x_1(nT)]+BR[x_2(nT)] &= AR[a^n x_1(nT-T)]+BR[a^n x_2(nT-T)] \\ &= Aa^n x_1(nT-T)+Ba^n x_2(nT-T) \\ \therefore R[Ax_1(nT)+Bx_2(nT)] &= AR[x_1(nT)]+BR[x_2(nT)] \end{aligned}$$

以上より、本システムは線形システムである。

◆時不変性の検証 (任意の時間遅れ k を想定)

$$\begin{aligned} y(nT) &= a^n x(nT-T) \quad (|a|<1) \\ y(nT-kT) &= a^{n-k} x(nT-kT-T) \\ R[x(nT-kT)] &= a^n x(nT-kT-T) \\ \therefore y(nT-kT) &\neq R[x(nT-kT)] \end{aligned}$$

以上より、本システムは時不変システムではない。

◆因果性の判定 (※このシステムは線形時不変システムではないので、因果性の定義に従い、「過去のサンプルだけを使って処理を行うか」をチェックする。)

$$\begin{aligned} y(nT) &= a^n x(nT-T) \quad (|a|<1) \\ n=-1 &\text{ とすると、} \\ y(-T) &= a^{-1} x(-T-T) \\ y(-T) &= a^{-1} x(-2T) \end{aligned}$$

以上より、時刻 $-T$ に対して過去の信号 x(-2T) しか参照しないため、定義を満たす。よって、このシステムは因果性システムである。

2.4.3 問題 2.1 (6)

まず、与式を展開する。

$$\begin{aligned} y(nT) &= \{an + x(nT+2T)\}^2 \\ &= a^2n^2 + 2anx(nT+2T) + x^2(nT+2T) \end{aligned}$$

◆線形性の検証 (任意の実定数入力 A,B を想定)

$$\begin{aligned} y(nT) &= a^2n^2 + 2anx(nT+2T) + x^2(nT+2T) \\ R[Ax_1(nT) + Bx_2(nT)] &= a^2n^2 + 2an[Ax_1(nT+2T) \\ &\quad +Bx_2(nT+2T)] + [Ax_1(nT+2T) + Bx_2(nT+2T)]^2 \\ &= a^2n^2 + 2anAx_1(nT+2T) + 2anBx_2(nT+2T) + A^2x_1^2(nT+2T) \\ &\quad +B^2x_2^2(nT+2T) + 2ABx_1(nT+2T)x_2(nT+2T) \\ AR[x_1(nT)] + BR[x_2(nT)] &= A(a^2n^2 + 2anx_1(nT+2T) \ldots \text{(以下略)} \\ \therefore R[Ax_1(nT) + Bx_2(nT)] &\neq AR[x_1(nT)] + BR[x_2(nT)] \end{aligned}$$

以上より、本システムは線形システムではない。

◆時不変性の検証 (任意の時間遅れ k を想定)
計算は省略するが、本システムは時不変システムではない。(x の関係しない n が存在する時点で、このシステムは時不変システムではない。)

◆因果性の判定 (※このシステムは線形時不変システムではないので、因果性の定義に従い、「過去のサンプルだけを使って処理を行うか」をチェックする。)

$$\begin{aligned} y(nT) &= a^2n^2 + 2anx(nT+2T) + x^2(nT+2T) \\ n = -1 \text{ とすると、} & \\ y(-T) &= a^2 - 2ax(-T+2T) + x^2(-T+2T) \\ y(-T) &= a^2 - 2ax(T) + x^2(T) \end{aligned}$$

以上より、時刻 $-T$ に対して未来の信号である $x(T)$ を必要とするので、定義を満たさない。よって、このシステムは因果性システムではない。

2.5 問題 2.2

回路を差分方程式で表すと以下のようになる。

(b) $y(nT) = 4x(nT) + 6x(nT - T) + x(nT - 2T)$

(f) $y(nT) = 5x(nT) + 14x(nT - T) + 11x(nT - 2T) + 6x(nT - 3T)$

問題 1.9 で作成したプログラムを少し改造し、それぞれのインパルス応答を求めた。

◆実行結果

```
----------------------- (b) -----------------------
n         0        1        2        3        4        5        6        7        8        9
y(nT) +4.0000  +6.0000  +1.0000  +0.0000  +0.0000  +0.0000  +0.0000  +0.0000  +0.0000  +0.0000
----------------------- (f) -----------------------
n         0        1        2        3        4        5        6        7        8        9
y(nT) +5.0000  +14.0000  +11.0000  +6.0000  +0.0000  +0.0000  +0.0000  +0.0000  +0.0000  +0.0000
```

◆ C 言語によるプログラム (p2.2.c)

```
// 2007.05.17(Thu)  0:4*  完成
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>

#define MAX 100
#define R   10

void output(double *dd) {
    int i;
    printf("n     "); for(i=0; i<R; i++) printf("%5d    ",i); puts("");
    printf("y(nT) "); for(i=0; i<R; i++) printf("%+1.4f  ",dd[i]); puts("");
    return;
}

int main() {
    int t,i;
    double x[MAX], y_b[MAX], y_f[MAX];

    // 入力信号 (インパルス)
    memset(x,0,MAX*sizeof(double));
    x[0] = 1;

    //*********************************************** (b)
    for(t=0; t<R; t++) {
        y_b[t] = 4.0*x[t];
        if(t>=1) y_b[t] += 6.0*x[t-1];
        if(t>=2) y_b[t] += x[t-2];
    }
    puts("----------------------- (b) -----------------------");
    output(y_b);

    //*********************************************** (f)
    for(t=0; t<R; t++) {
        y_f[t] = 5.0*x[t];
        if(t>=1) y_f[t] += 14.0*x[t-1];
        if(t>=2) y_f[t] += 11.0*x[t-2];
        if(t>=3) y_f[t] += 6.0*x[t-3];
    }
    puts("----------------------- (f) -----------------------");
    output(y_f);

    return 0;
}
```

055762F 本村健太 ／ 2007.05.17(Tue) AM1:10
with LaTeX, Scilab & OmniGraffle