Inlämnat 20.11.67
Godk. 24.11.67

En metod för tidsdomänsyntes.

Examensarbete av Staffan Kjerrström

1967

Innehåll.

Sid

Referenser.

Vasiliu C.G.; A practical method for time-domain network synthesis,
IEE Trans. on Circuit Theory vol 2, June 1965. [1]

Kendall; Active network synthesis [2]

Baldetorp P.; Aktiva RC-nät [3]

Hazony-Lagerlöf; Canonical active networks,
Chalmers, 1966 [4]

Zetterberg, Salomonsson, Gunmar;
Dataöverföring för undervisning, Rapport nr 1,
LTH 1965 [5]

Yengst; Procedures of modern network synthesis,
kap 15. Approximation in time domain.

Storer; Passive network synthesis.

1. Inledning.

Syntesen av ett elektriskt nät med önskade egenskaper kan uppdelas i två etapper. För det första gäller det att bestämma nätets överföringsfunktion, och därefter att realisera nätet. Man måste ställa vissa krav på överföringsfunktionen, och dessa är beroende av hur man ämnar realisera nätet.

Om nätets egenskaper är givna i frekvensplanet så kan man bestämma överföringsfunktionen, $H(s)$, exakt eller approximativt, direkt ur de givna önskemålen.

Syntes i tidsdomänen förutsätter att egenskaperna är givna i form av ett önskat impulssvar $f(t)$ till nätet. Man söker sedan en realiserbar överföringsfunktion $H(s)$, vars impulssvar $h(t)$ i någon mening approximerar $f(t)$.

2. Problemställning.

Avsikten med detta examensarbete är att, med utgångspunkt ifrån en given metod för tidsdomänsyntes, konstruera ett Algolprogram för de beräkningar, som leder fram till överföringsfunktionen $H(s)$. Därefter diskuteras några metoder för realisering av $H(s)$ som ett passivt eller aktivt nät.

3. Beräkning av överföringsfunktionen.

3.1. Vald metod.

Ett flertal metoder för syntes i tidsdomänen finns utvecklade. De kan i stort indelas i två grupper. Den första består av metoder som fordrar att man känner, eller beräknar, det analytiska uttrycket för det önskade impulssvaret $f(t)$. $f(t)$ serieutvecklas i ett ortogonalt funktionssystem och serien trunkeras efter ett lämpligt antal termer. Dessa metoder har fördelen att man lätt kan beräkna felet i approximationen, till exempel kvadratiska medelfelet. En nackdel är att koefficienterna i serieutvecklingen är svåra att bestämma, särskilt om $f(t)$ endast är given i ett begränsat antal punkter. Den andra gruppen består av metoder som utnyttjar ett ändligt antal värden $f(t_i)$. Här blir felet större och mera svårkontrollerat.

Metoden som här valts är angiven av C.G. Vasiliu, [1]. Den använder sig av ett ändligt antal värden $f(t_i)$. Tack vare metodens systematiska karaktär är den väl lämpad för programmering. Dessutom får man, med programmets hjälp, god kontroll över felet.

I programmet består felkriteriet av den maximala skillnaden mellan $f(t)$ och $h(t)$ i gitterpunkterna t_i.

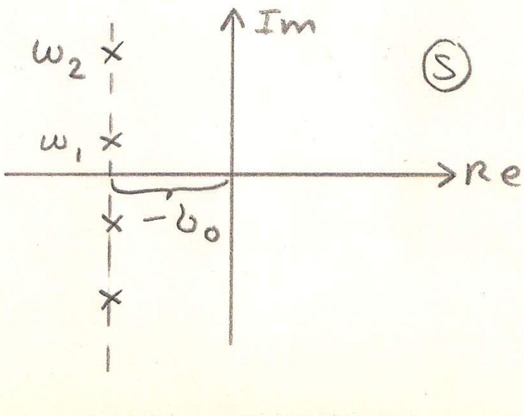
3.2. Beskrivning av metoden.

Vi antar att Laplacetransformen till $h(t)$, $H(s)$, är en realiserbar överföringsfunktion till ett passivt nät med diskreta element. Detta innebär att $H(s)$ får formen:

$$H(s) = \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0}{s^m + b_{m-1} s^{m-1} + \dots + b_0}$$

där $n \le m$ och a_ν och b_ν reella.
Dessutom måste polerna till $H(s)$ tillhöra vänstra halvplanet av s-planet.
Antag vidare att $h(t)$ ej innehåller en impuls i origo. Detta medför att $n < m$.

För att säkra nätets stabilitet bestämmer vi pollägena i förväg. De skall ligga på en linje parallell med imaginära axeln och med konstant avstånd från varandra, se fig. 1. Detta val förenklar dessutom de fortsatta räkningarna avsevärt.



Figur 1

Detta medför att $H(s)$ får formen,

$$H(s) = \sum_{\alpha} \frac{A_\alpha s + B_\alpha}{(s-b_0)^2 + \omega_\alpha}$$

där $b_0 < 0$ och $\omega_\alpha - \omega_{\alpha-1}$ är konstant.

Funktionen $f(t)$, som skall approximeras, antas vara skild från noll endast i intervallet $0 \leq t < t_0$. Detta intervall delas i n st delar med längden T, och dessa delas i sin tur i m st delar, se fig. 2.

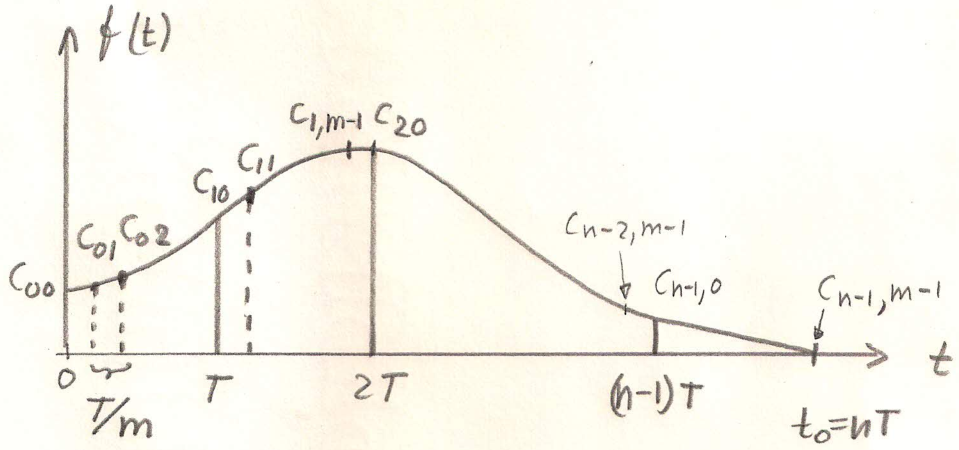


Fig. 2

$c_{i\eta}$ betecknar $f(t)$'s värden i dessa tidpunkter.

Alltså $c_{i,\eta} = f[(i + \eta/m)T]$.

Vid härledning av metoden användes den "avancerade" z-transformen, se [1], som utvecklas i potensserie i $\frac{1}{z}$. Koefficienterna framför z^{-i} ger då värdena på motsvarande impulssvar $h[(i+\eta/m)T]$.
$h[(i+\eta/m)T]$ antas ha värdena $c_{i\eta}$.

På detta sätt kommer $h(t)$ att approximera $f(t)$. Man får ett ekvationssystem med $n \cdot m$ ekvationer och n obekanta. De obekanta storheterna utgöres av koefficienterna i potensserien. Tack vare valet av pollägen hos $H(s)$ så blir kolonnvektorerna i ekvationssystemet ortogonala.

Genom att minimera det kvadratiska medelfelet, mellan vänster och höger led i ekvationssystemet, kan man beräkna optimala värden på de n obekanta. Med hjälp av tabeller för den inversa z-transformen beräknas, den till z-transformen hörande, Laplacetransformen $H(s)$. Följande resultat erhålles:

$$h(t) = b^{t/T} \sum_{\alpha} \left(M_\alpha \cos \omega_\alpha t + N_\alpha \sin \omega_\alpha t \right)$$

$$H(s) = \sum_{\alpha} \frac{M_\alpha (s - b_0) + N_\alpha \omega_\alpha}{(s - b_0)^2 + \omega_\alpha^2}$$

$$\omega_\alpha = \begin{cases} \dfrac{\pi}{nT}(2\alpha - 1) & \text{då} \quad b_0 < 0 \\[2ex] \dfrac{2\pi}{nT}\alpha & \text{då} \quad b_0 > 0 \end{cases}$$

$$b = \sqrt[n]{|b_0|} \qquad b_0 = \frac{\ln b}{T} \qquad |b_0| < 1$$

Fall 1 $\quad b_0$ negativ

$$M_\alpha = \frac{2}{mnb_0} \sum_{\eta=0}^{m-1} \sum_{k=1}^{n} b^{k-\eta/m} \cdot C_{n-k,\eta} \cos\left[\left(k-\frac{\eta}{m}\right)(2\alpha-1)\frac{\pi}{n}\right]$$

$$N_\alpha = \frac{-2}{mnb_0} \sum_{\eta=0}^{m-1} \sum_{k=1}^{n} b^{k-\eta/m} \cdot C_{n-k,\eta} \sin\left[\left(k-\frac{\eta}{m}\right)(2\alpha-1)\frac{\pi}{n}\right]$$

$$\alpha = \begin{cases} 1,2,\dots \frac{n}{2} & n \text{ jämnt} \\ 1,2,\dots \frac{n+1}{2} & n \text{ udda} \end{cases}$$

Fall 2 $\quad b_0$ positiv

$$M_\alpha = \frac{2}{mnb_0} \sum_{\eta=0}^{m-1} \sum_{k=1}^{n} b^{k-\eta/m} C_{n-k,\eta} \cos\left[\left(k-\eta/m\right)\frac{2\pi}{n}\alpha\right]$$

$$N_\alpha = \frac{-2}{mnb_0} \sum_{\eta=0}^{m-1} \sum_{k=1}^{n} b^{k-\eta/m} C_{n-k,\eta} \sin\left[\left(k-\frac{\eta}{m}\right)\frac{2\pi}{n}\alpha\right]$$

$$\alpha = \begin{cases} 0,1,2,\dots \frac{n-1}{2} & n \text{ udda} \\ 0,1,2,\dots \frac{n}{2} & n \text{ jämnt} \end{cases}$$

Anmärkning $\quad$ Om b_0 är negativ och n udda så skall endast halva värdet av $M_{\frac{n+1}{2}}$ resp. $N_{\frac{n+1}{2}}$ användas vid beräkning av $h(t)$ i första formeln. Om b_0 är positiv så skall endast halva värdet av M_0, $M_{\frac{n}{2}}$ och $N_{n/2}$ användas vid beräkning av $h(t)$.

Här är b_0 en koefficient i z-transformen och den svarar emot polernas avstånd från imaginära axeln på följande sätt:

$$|b_0| = \exp \delta_0 nT .$$

Eftersom $\delta_0 < 0$ så gäller att $|b_0| < 1$. Observera att om man väljer b_0 positiv och n jämnt eller b_0 negativ och n udda så blir överförings-funktionens gradtal $n+1$ och inte n.

Om man väljer $m=1$ så har ekvations-systemet en unik lösning, och $h(t)$ samman-faller med $f(t)$ i tidpunkterna $t_i = i \cdot T$ där $i = 0, 1, 2, \ldots n-1$. Detta kan användas för samplade datasystem då man önskar att $h(t)$ skall anta givna värden vid sampeltid-punkterna.

3.3. Approximationsfel.

Den givna funktionen $f(t)$ har antagits vara noll utanför intervallet $(0, t_0)$. Sätt $h(t) = h_1(t) + e(t)$, där $h(t)$ representeras av $h_1(t)$ i intervallet $(0, t_0)$ och av $e(t)$ för $t > t_0$. Då blir

$$e(t) = \sum_{k=1}^{\infty} b_0^k \, h_1(t - k t_0) .$$

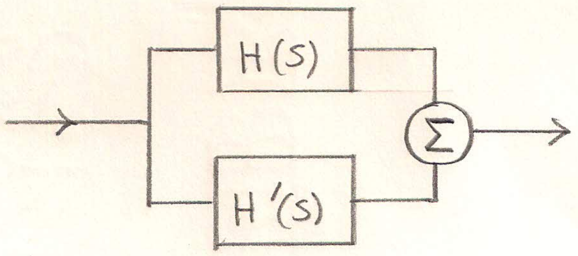
det vill säga, utanför $(0, t_0)$ uppträder "ekon" av $h_1(t)$ med minskande amplitud.

sätt $\dfrac{\max |e(t)|}{\max |h_1(t)|} = E$

det vill säga E är ett mått på felet för $t > t_0$.
Då blir $E = |b_0|$.

Vid minimeringen av kvadratiska medelfelet användes ett vägt medelvärde med vikten $|b_0|^{1/n} - t/T$. Om man alltså minskar b_0 för att få litet fel för $t > t_0$ så tenderar felet $|f(t) - h(t)|$ att öka för små t-värden.* Detta kan kompenseras genom att öka n, dvs gradtalet på överföringsfunktionen $H(s)$. Även genom att öka finindelningen av tidsintervallen, m, minskar man felet. m har dock mindre inverkan på felet, se vidare under "Resultat av programkörningar."

Felet för $t > t_0$ kan minskas genom följande koppling.



Här är $H'(s)$ den överföringsfunktion som man får genom att ge b_0 motsatt tecken mot det som användes för $H(s)$. På detta sätt kommer de udda "ekona" att få motsatt tecken och eleminera varandra. Då blir $E = b_0^2$.

* Jämför exempel 2 sid 33 och ex 15 sid 38.

4. Programmet.

4.1. Beskrivning.

Programmet är utformat för att köras vid Lunds datacentral och alltså skrivet på s.k. Smil-Algol. Det följer Algol 60, utom vad beträffar förekomsten av bokstäverna å, ä och ö i kommentarerna. Dessutom användes Smil-Algols konventioner för in och utmatning.

Följande storheter beräknas av programmet.

1) Koefficienterna $a_0, a_1 \ldots a_n$ och $b_0, b_1 \ldots b_m$ till $H(s)$ på formen:

$$H(s) = \frac{a_0 + a_1 s + \cdots + a_n s^n}{b_0 + b_1 s + \cdots + b_m s^m}$$

2) b och δ_0 .

3) M_α, N_α, ω_α och ω_α^2 för alla aktuella α värden.

$$H(s) = \sum_\alpha \frac{M_\alpha (s-\delta_0) + N_\alpha \omega_\alpha}{(s-\delta_0)^2 + \omega_\alpha^2} .$$

4) Resulterande impulssvar $h(t)$ i gitterpunkterna:

$$h[(f + g/m)T] \qquad \begin{cases} f = 0, 1, 2 \ldots 2 \times n \\ g = 0, 1, 2 \ldots m-1 \end{cases}$$

5) Felet $e(t) = h(t) - f(t)$ i gitter-punkterna:

$$e[(f + g/m)T] \qquad \begin{cases} f = 0, 1, 2 \ldots n-1 \\ g = 0, 1, 2 \ldots m-1 \end{cases}$$

Programmet består av tre delar, varav del 1 och 3 alltid användes, medan del 2 endast användes om det önskade impuls-svaret $f(t)$ är givet i analytisk form. Man får då, som del 2, skriva en procedur för $f(t)$.

Första delen består av ett blockhuvud med deklaration av använda variabler och procedurer. Beteckningarna ansluter så långt som möjligt till de som använts vid beskrivningen av metoden.

Följande ändringar och tillägg är gjorda:

Beteckning	I programmet
α	a
i	f
η	g
b_0	so
π	pi
M_α	M[a]
N_α	N[a]
ω_α	W[a]
$c_{i\eta}$	C[f, g]

Övriga storheter har oförändrade beteckningar, eller används endast inom programmet.

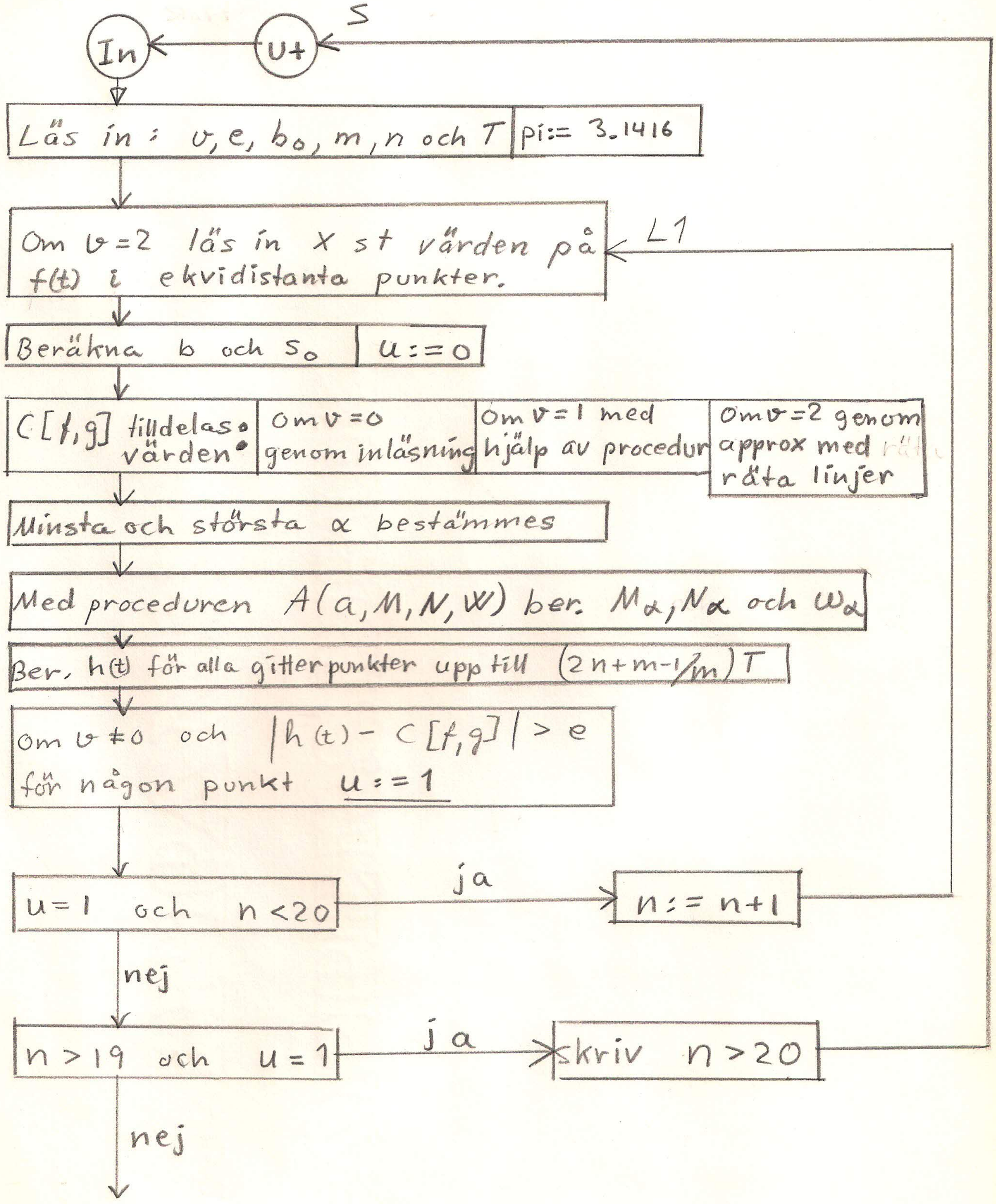
Den första proceduren, h(t), användes för att beräkna det approximativa impulssvaret h(t) för en given tidpunkt t.

$$h(t) = b^{t/\tau} \sum_{\alpha} \left(M_\alpha \cos \omega_\alpha t + N_\alpha \sin \omega_\alpha t \right).$$

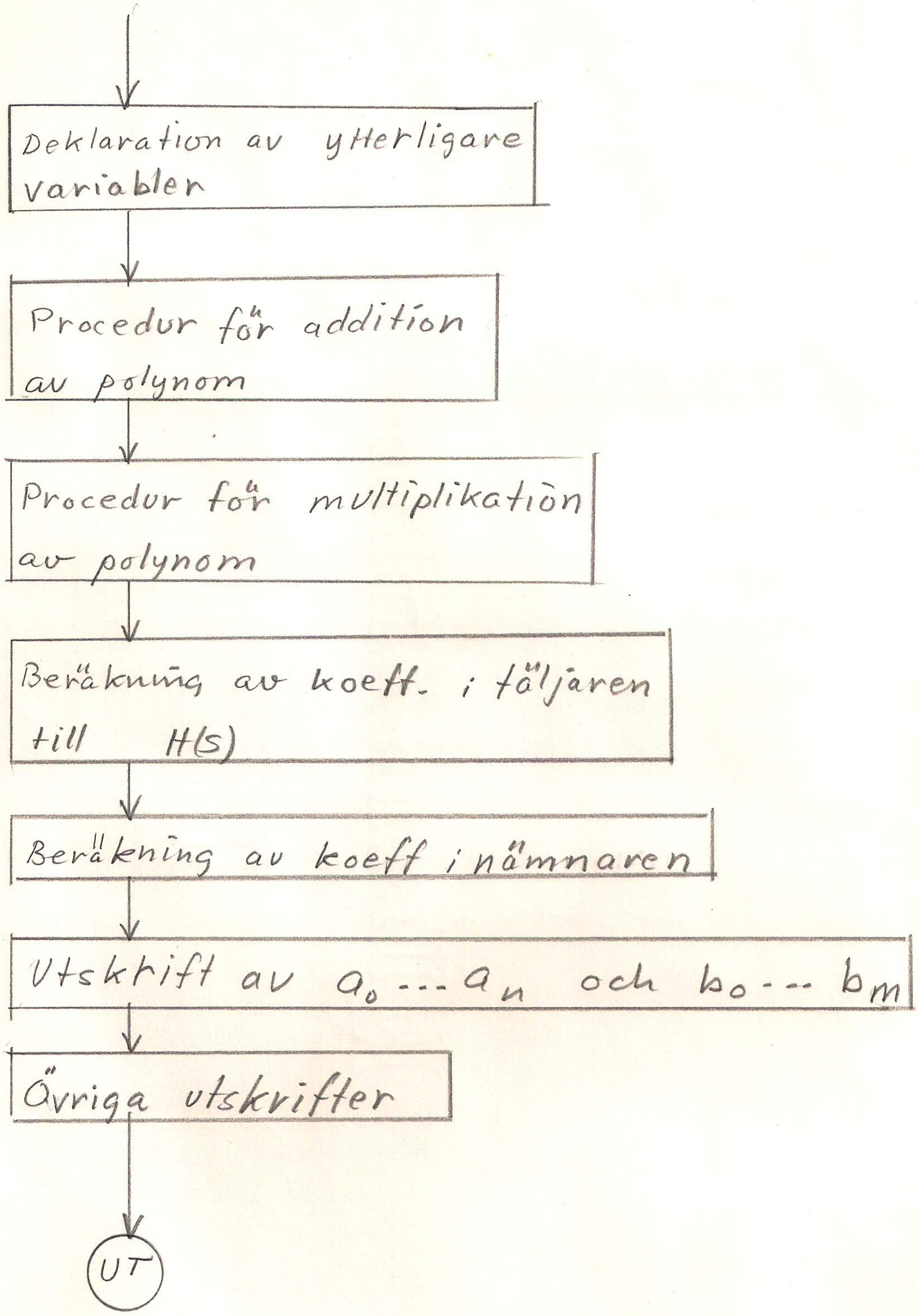
Nästa procedur, A(a, M, N, W), beräknar koefficienterna M_α, N_α och ω_α för ett givet $\alpha = a$. Se formler på sid 7 och 8.

Programmets andra del består av en procedur för f(t) och användes vid beräkningsalternativ 2, se "Användning av programmet."

Sista delen av programmet består av inläsning av data, beräkningar och utskrift, enligt följande förenklade schema:



vänd


Deklaration av ytterligare variabler
Procedur för addition av polynom
Procedur för multiplikation av polynom
Beräkning av koeff. i täljaren till H(s)
Beräkning av koeff i nämnaren
Utskrift av $a_0 \cdots a_n$ och $b_0 \cdots b_m$
Övriga utskrifter
UT

```
S:begin comment Del 1 av program för beräkningar enligt

        Vasiliu: A practical method for time-domain network synthesis,

        IEEE Circuit theory number 2 June 1965;

        integer a,d,f,g,m,n,u,v,x,y; real b,b0,e,s0,T,pi,r;
        array M,N,W[0:10],C[0:20,0:14],H[0:40,0:14],X[0:50];


        real procedure h(t); value t; real t;
        begin real S; integer a; S:=0;
            for a:=1 step 1 until (n-1)/2 do
            S:=S+M[a]×cos(W[a]×t)+N[a]×sin(W[a]×t);
            if b0<0^entier(n/2)=n/2then
            S:=S+M[n/2]×cos(W[n/2]×t)+N[n/2]×sin(W[n/2]×t);
            if b0<0^entier(n/2)≠n/2then
            S:=S+M[(n+1)/2]×cos(W[(n+1)/2]×t)/2+N[(n+1)/2]×sin(W[(n+1)/2]×t)/2;
            if b0>0then S:=S+M[0]×cos(W[0]×t)/2+N[0]×sin(W[0]×t);
            if b0>0^entier(n/2)=n/2then
            S:=S+M[n/2]×cos(W[n/2]×t)/2+N[n/2]×sin(W[n/2]×t)/2;
            h:=b↑(t/T)×S
        end;


        procedure A(a,M,N,W); comment Beräknar koefficienterna M,N och W
                              för visst alfa=a;
        integer a; real M,N,W;
        begin integer k,l;real p,q,S,U;
            if b0<0 then p:=(2×a-1)×pi/n else p:=2×a×pi/n;  S:=U:=0;
            for l:=0 step 1 until m-1 do for k:=1 step 1 until n do
              begin q:=b↑(k-l/m)×C[n-k,l]; S:=S+q×cos((k-l/m)×p);
                  U:=U+q×sin((k-l/m)×p)
              end;
            q:=m×n×b0; M:=2×S/q; N:=(-2)×U/q; W:=p/T
        end;


        comment Här inläses ev. en procedur för det önskade impulssvaret
                F(t),som del 2;;
```

```
begin comment Del 3;
    v:=read; e:=read; b0:=read; m:=read; n:=read; T:=read;  pi:=3.1416;

    if v=2 then begin x:=read; for y:=0 step 1 until x do
                      X[y]:=read
                end;

  L1: b:=abs(b0)↑(1/n); s0:=ln(b)/T; u:=0;
    for f:=0 step 1 until n-1 do
    for g:=0 step 1 until m-1 do
    if v=1 then C[f,g]:=F((f+g/m)×T) else
    if v=2 then begin y:=entier(x×(f+g/m)/n); C[f,g]:=(x×(f+g/m)/n-y)×X[y+1]+
                      (1+y-x×(f+g/m)/n)×X[y]
                end
            else C[f,g]:=read;
    if entier(n/2)=n/2 then d:=n/2 else
    if b0<0 then d:=(n+1)/2 else d:=(n-1)/2;
    for a:=if b0<0 then 1 else 0 step 1 until d do
        A(a,M[a],N[a],W[a]);
    for f:=0 step 1 until 2×n do
    for g:=0 step 1until m-1 do
        begin H[f,g]:=h((f+g/m)×T);
        if v≠0∧f<n then begin if abs(H[f,g]-C[f,g])>e then u:=1 end
    end;
    if u=1∧n<20 then  begin n:=n+1; go to L1 end;

    if n>19∧u=1 then
    begin punch(1); write(⊲ N.>.20⊳); go to S end;

    begin comment Beräkning av H(s) med gemensam nämnare;
        integer a1,b1,c1,d1,e1; array A,B,C,D,E[0:42];
        procedure Add(X,Y,Z); array X,Y,Z;
            begin for d1:=0 step 1 until n+1 do Z[d1]:=X[d1]+Y[d1]
            end;
        procedure Mult(X,Y,Z); array X,Y,Z;
            begin for d1:=0 step 1 until n+1 do Z[d1]:=0;
            for d1:=0 step 1 until n+1 do for e1:=0 step 1 until n+1 do
            Z[d1+e1]:=Z[d1+e1]+X[d1]×Y[e1]
            end;
    for b1:=2 step 1 until n+1 do begin A[b1]:=0; E[b1]:=0 end;

    A[1]:=-2×s0; A[2]:=1; c1:=if b0<0 then 1 else 0;
        for a1:=0 step 1 until n+1 do D[a1]:=0;
        for a1:=c1 step 1 until d do
        begin B[0]:=1; for d1:=1 step 1 until n+1 do B[d1]:=0;
            for  b1:=c1 step 1 until d do if a1≠b1 then
            begin A[0]:=W[b1]↑2+s0↑2; Mult(A,B,C);
                for d1:=0 step 1 until n+1 do B[d1]:=C[d1];
            end;
            E[0]:=N[a1]×W[a1]-M[a1]×s0; E[1]:=M[a1];
```

```
Mult(E,B,C); Add(C,D,D)
end;

        B[0]:=1; for a1:=1 step 1 until n+1 do B[a1]:=0;
        for a1:=c1 step 1 until d do
        begin A[0]:=W[a1]↑2+s0↑2; Mult(A,B,C);
            for b1:=0 step 1 until n+1 do B[b1]:=C[b1]
        end;


        write(⊲A0␣␣␣␣␣␣␣␣␣␣␣␣A1␣␣␣␣␣␣␣␣␣␣␣␣␣A2␣␣...⊳); punch(1);
        for a1:=0 step 1 until n+1 do
        begin print(5,5,D[a1]); if a1=5∨a1=11∨a1=17 then punch(1) end;
        punch(1); punch(1);
        write(⊲B0␣␣␣␣␣␣␣␣␣␣␣␣B1␣␣␣␣␣␣␣␣␣␣␣␣␣B2␣␣...⊳); punch(1);
        for b1:=0 step 1 until n+1 do
        begin print(5,5,B[b1]); if b1=5∨b1=11∨b1=17 then punch(1) end;
        punch(1); punch(1)
end;
     write(⊲alfa␣␣M␣␣␣␣␣␣␣␣␣␣␣␣␣N␣␣␣␣␣␣␣␣␣␣␣␣␣W␣␣␣␣␣␣␣␣␣␣␣␣␣W↑2⊳);
     for a:=if b0<0 then 1 else 0 step 1 until d do
     begin punch(1); print(1,0,a); print(5,5,M[a]); print(N[a]); print(W[a]);
         r:=W[a]↑2; print(r)
     end;
     punch(1); punch(1); punch(1); print(2,0,v); print(5,5,e); print(5,5,b0);
     print(4,0,m); print(n); print(5,5,T); punch(1); punch(1);
     write(⊲b␣resp.␣sigma0␣=⊳); print(2,5,b); print(5,5,s0);
     punch(15); space(3);
     for g:=0 step 1 until m-1 do begin print(2,0,g); space(6) end; punch(1);
     for f:=0 step 1 until 2×n do
     begin punch(1); print(2,0,f);
         for g:=0 step 1 until m-1 do print(2,5,H[f,g])
     end;
     punch(1); punch(1);
     for f:=0 step 1 until n-1 do
     begin punch(1); print(2,0,f);
         for g:=0 step 1 until m-1 do print(2,5,H[f,g]-C[f,g])
     end;
      go to S
 end
 end
```

4.2. Användning av programmet.

De data som behövs skrives enligt Algol-60 på 5-kanals hålremsa. På remsan skall talen stå i följande ordning:

v, e, b_0, m, n, T. Därefter kommer värden på $f(t)$ enligt alternativen nedan.

v betecknar vilket alternativ som används.
e betecknar maximalt tillåtet fel i intervallet $(0, nT)$.

T väljes lämpligen lika med 1. Om T väljes för stor så minskar noggrannheten vid beräkning av w_α och om T väljes för liten så kan man få fel i utskriften. När man sedan återgår till onormerad tid så skall man dividera w_α och b_0 med T.
Om man väljer $\max |f(t)| = 1$ så blir det relativa felet E lika med absoluta felet. Multiplikation av $f(t)$ med en konstant motsvaras av multiplikation av $H(s)$ med samma konstant.

Beräkningarna kan göras enligt följande tre alternativ:

<u>Alt 1.</u> Programmet genomlöpes endast en gång med alla givna parametrar fixa. $f(t)$ inmatas i form av värden i de använda punkterna $C[f, g] = f[(f + g/m)T]$

$f = 0, 1, 2 \ldots n-1 \qquad g = 0, 1, 2, \ldots m-1$

På dataremsan skall stå:

0   1 b_0 m   n   T   C[0,0]   C[0,1] ---
--- C[0, m-1]   C[1,0] ----   C[n-1, m-1]

Andra talet på remsan, d.v.s. e, kan här väljas godtyckligt.

<u>Alt 2</u>. f(t) ges i form av en procedur, vilken inläses som del 2 av programmet.
Proceduren skall börja på följande sätt:
<u>real</u> <u>procedure</u> F(t) ; <u>real</u> t ;

och avslutas lämpligen med två semikolon (för inläsningen).
Vidare skall proceduren skrivas med storheten n·T som tidsenhet, så att $t_0 = nT$ är fix i förhållande till funktionen f(t), oberoende av n.
På dataremsan skall stå: 1   e b_0 m   n   T

Välj e som det maximalt tillåtna felet i intervallet (0, nT).

$$e = \max |h(t_i) - f(t_i)|.$$

Observera att programmet beräknar e som max e(t) i de använda gitterpunkterna. Om m=1 så blir e(t) noll i gitterpunkterna och däremellan har man ingen kontroll på felet.

b_0 väljes som det maximala felet för $t > nT$.

$$b_0 = \pm \frac{\max |e(t)|}{\max |h(t)|} \quad ;$$

Programmet börjar med givet startvärde på n och om felet e blir för stort så ökas n, ett steg i taget, tills felet blir tillräckligt litet. Om n blir större än 20 så utskrives detta och maskinen stannar.

Alt 3. Önskat impulssvar ges i form av ett lämpligt antal ekvidistanta värden.

$$f(i \cdot t_1) \qquad i = 0\ 1, 2, \dots X \qquad X \leq 50$$

Punkten $t = nT$ kommer då att motsvaras av $i = X$. Beräkningarna utföres på samma sätt som i alt 2, så att n ökas ett steg i taget tills felet underskrider e.

På dataremsan skall stå:

$$2 \quad e \quad b_0 \quad m \quad n \quad T \quad X \quad f(0) \quad f(t_1)$$
$$f(2t_1) \quad \dots \quad f(X t_1)$$

Man bör välja $|b_0|$ så stor som möjligt med hänsyn till felet i $t > nT$. Då får man ju lägst gradtal på överföringsfunktionen för givet e. Dock gäller att $|b_0| < 1$.

För närmare diskussion om läget av punkten $t_0 = nT$, i förhållande till $f(t)$ hänvisas till ref[5] sid 29–31. Sammanfattningsvis kan sägas att $t_0 = nT$ bör förläggas i närheten av det största t-värde för vilket $f(t)$ antas skild från noll.

4.3. Begränsningar.

Av metoden betingade begränsningar är bland annat följande:

$|b_0| < 1$, $\vartheta \in \{0, 1, 2\}$ och m och n naturliga tal

I programmet införda begränsningar på grund av resultatutskriften:

$m \leq 12$ $\quad 2 \leq n \leq 20 \quad |f(t)| <$ ca 50

Det är praktiskt att välja $\max |f(t)| = 1$

4.4. Utskrift.

Resultatet utskrives på följande form:

$A_0 \quad A_1 \quad A_2 \; ---$

$a_0 \quad a_1 \quad a_2 \; ---- \; a_n$

$B_0 \quad B_1 \quad B_2 \; ---$

$b_0 \quad b_1 \quad b_2 \; ---- \; b_m$

ALFA	M	N	W	W↑2
0	M_0	N_0	w_0	w_0^2
1	M_1	N_1	w_1	w_1^2
2	M_2	N_2	w_2	w_2^2
⋮	⋮	⋮	⋮	⋮

$\vartheta \quad e \quad b_0 \quad m \quad n \quad T$

B Resp SIGMA O = $b \quad \sigma_0$

<u>ny sida</u> →

$\begin{array}{c|c} & 0 \quad 1 \; ---- \; m-1 \\ 0 & \\ \vdots & \text{tabell över } h((\ell + \nu/m)T) \\ 2n & \end{array}$

$\begin{array}{c|c} & 0 \quad 1 \; ---- \; m-1 \\ 0 & \\ \vdots & \text{tabell över } e(t) = h(t) - f(t) \\ n-1 & \end{array}$

5. Resultat av programkörningar

Under arbetet med programmets utformning har följande tre exempel körts, med konstant gradtal på $H(s)$ och utan att beräkna konstanterna $A_0\ A_1\ \dots\ B_0\ B_1\ \dots$

Önskat impulssvar $= f(t)$.

1) $f(t)$ likbent triangel $b_0 = 0{,}1$ $m = 2$ $n = 5$

Resultat: se diagram, sid 32 och utskrift 1, sid 39.

Maximalt fel i intervallet $(0, nT)$

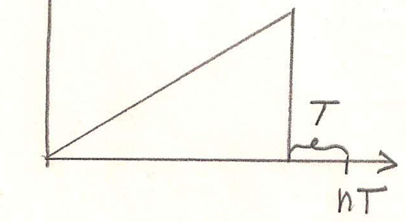
$e_{max} = 0{,}079$

2) $f(t)$ rektangel $b_0 = 0{,}05$ $m = 2$ $n = 8$

Resultat: Diagram, sid 33 och utskrift 2, sid 40.

$e_{max} = 0{,}133$

3) Med variabelt n



Max tillåtet fel i $(0, nT)$

$e = 0{,}05$

$b_0 = 0{,}05$

$m = 4$

Resultat $n > 20$, se utskrift 3, sid 41.

De två första exemplen är valda ur ref [1] för att möjliggöra en jämförelse av resultaten. Man finner att dessa överensstämmer.

För att testa det färdiga programmet och för att belysa följande frågor gjordes en serie körningar där f(t) valts som $\sin^2\left(\pi \times \frac{t}{0{,}9nT}\right) \quad 0 \le t \le 0{,}9nT$

a) Hur påverkar m, d.v.s. finindelningen av intervallet $(0, nT)$, det maximala felet i samma intervall?

b) Är läget på punkten $t = nT$ i förhållande till f(t) kritiskt?

f(t) gavs dels i form av en procedur, se nästa sida, och dels i form av värden i ekvidistanta punkter för exempel 12, 13 och 14.

```
real procedure F(t); real t;
    begin if t<0.9×n×T then F:= sin(pi×t/0.9/n/T)↑2 else F:=0
    end;;
```

Indata för ex 4 – 14

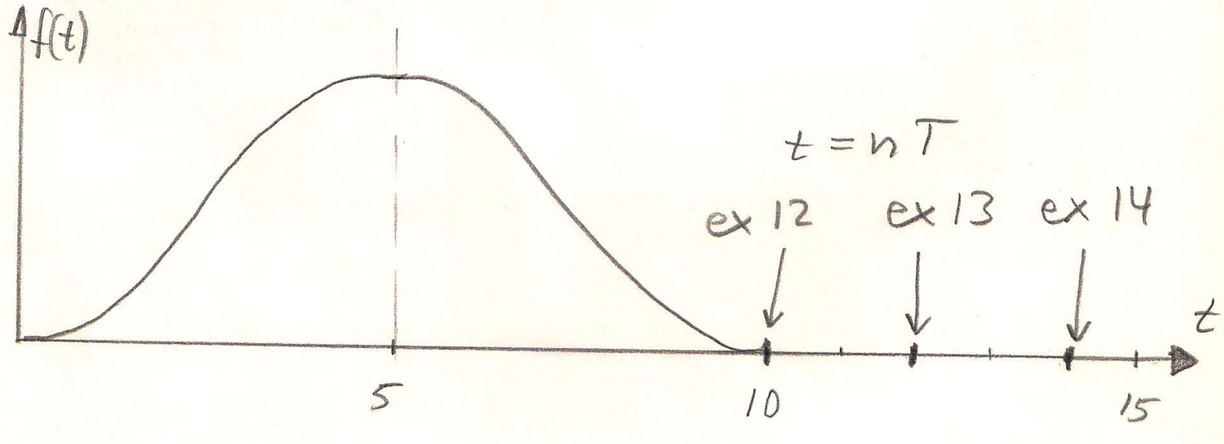
	n-konst				e konst.				Var. av nT		
Ex nr	4	5	6	7	8	9	10	11	12	13	14
ϑ	1	1	1	1	1	1	1	1	2	2	2
e	1	1	1	1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
b_0	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
m	2	4	6	10	2	4	6	10	4	4	4
n	8	8	8	8	4	4	3	3	4	4	4
T	1	1	1	1	1	1	1	1	1	1	1
X =									10	12	14
									0		
									0,0955		
									0,3457		
									0,6540		
Tabell över f(t)									0,9044	=	=
									1		
									0,9044		
									0,6540		
									0,3457		
									0,0955		
									0	0	0
										0	0
											0
											0

I ex. 4–7 varieras m från 2 till 10 vid konstant gradtal n = 8 för att studera inverkan på felet i (0, nT).
Man finner att ändringen av felet i gitterpunkterna blir liten och till synes slumpmässig, se sid 42–45. Detta kan bero på att felet ändras mindre än 10^{-5} eller att felet minskar endast mellan gitter-

punkterna. Det senare kan ej undersökas med hjälp av programmet. Diagram över felet för $m=2$ och $m=10$, se sid 34.
$h(t)$ för ex. 4, se sid 35.

I exempel 8–11 varieras m från 2 till 10 med givet max. fel i $(0, nT)$ $e_{max}=0,1$. n är här variabelt för att undersöka om man genom att öka m, kan minska gradtalet n vid konstant e_{max}.
Resultat, se sid 46–49. I samtliga fall erhålles $n=5$ dock minskar $e(t)$ något med m, se diagram sid 36. Minskningen är märkbar endast då m ökar från 2 till 4.

I exempel 12–14 varieras $t=nT$ i förhållande till $f(t)$ enligt figur. Dessa exempel avser också att kontrollera att programmet fungerar enligt alternativ 3 och därför blir $f(t)$ approximerad med räta linjer.



Vid alla tre exemplen användes $e_{max}=0,1$, $m=4$ och variabelt n.

Resultat:

Ex	12	13	14
n	5	5	4
max $e(t)$	0,096	0,049	0,090

Se sid 50–52. Här inverkar alltså nT:s läge på n. Då avståndet till nT ökas så kommer det maximala felet för $t > nT$ att flyttas mot större t. Felet $e(t)$ för ex 12 och 14, se diagram på sid 37.

Efter denna körning upptäcktes fel i programmet som påverkar beräkningen av $A_0 \cdots A_n$ och $B_0 \cdots B_m$. Dessa värden har därför utelämnats i utskrifterna.

För att kontrollera beräkningen av $A_0 \cdots A_n$ och $B_0 \cdots B_m$, efter rättning av programmet, kördes följande exempel:

Ex 15 $f(t)$ rektangelpuls, se nästa sida.

$e = 0,1$ $b_0 = 0,1$ $m = 2$ $n_0 = 3$

Resultat: $n = 8$ $e_{max}(t) = 0,09890$

$h(t)$, se diagram sid 38.

Utskrift se sid 53 och 54.

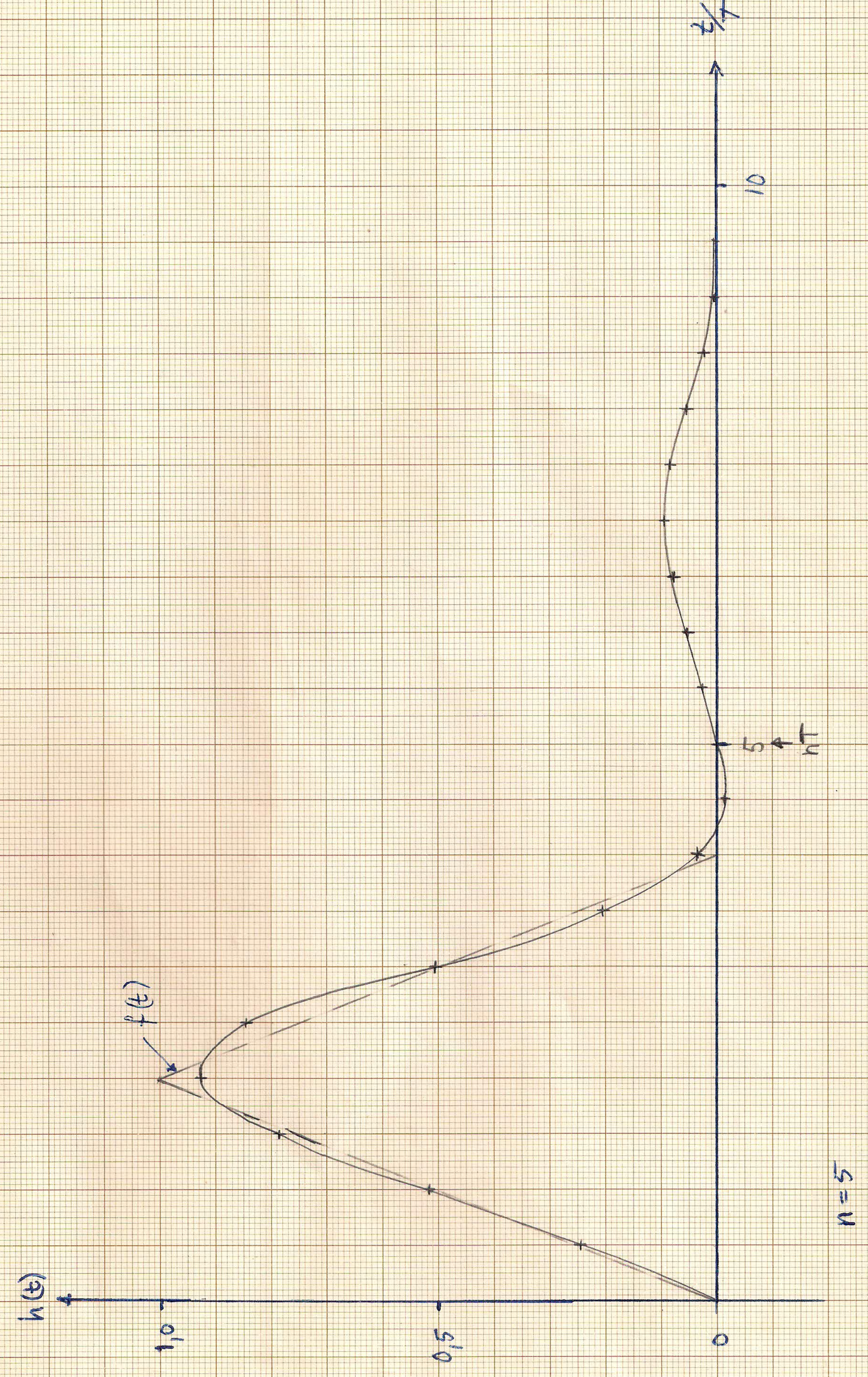
```
real procedure F(t); real t;

begin if t<3×n×T/4 then F:=1 else if t>3×n×T/4 then F:=0 else F:=0.5
end;;
```

Slutsatser.

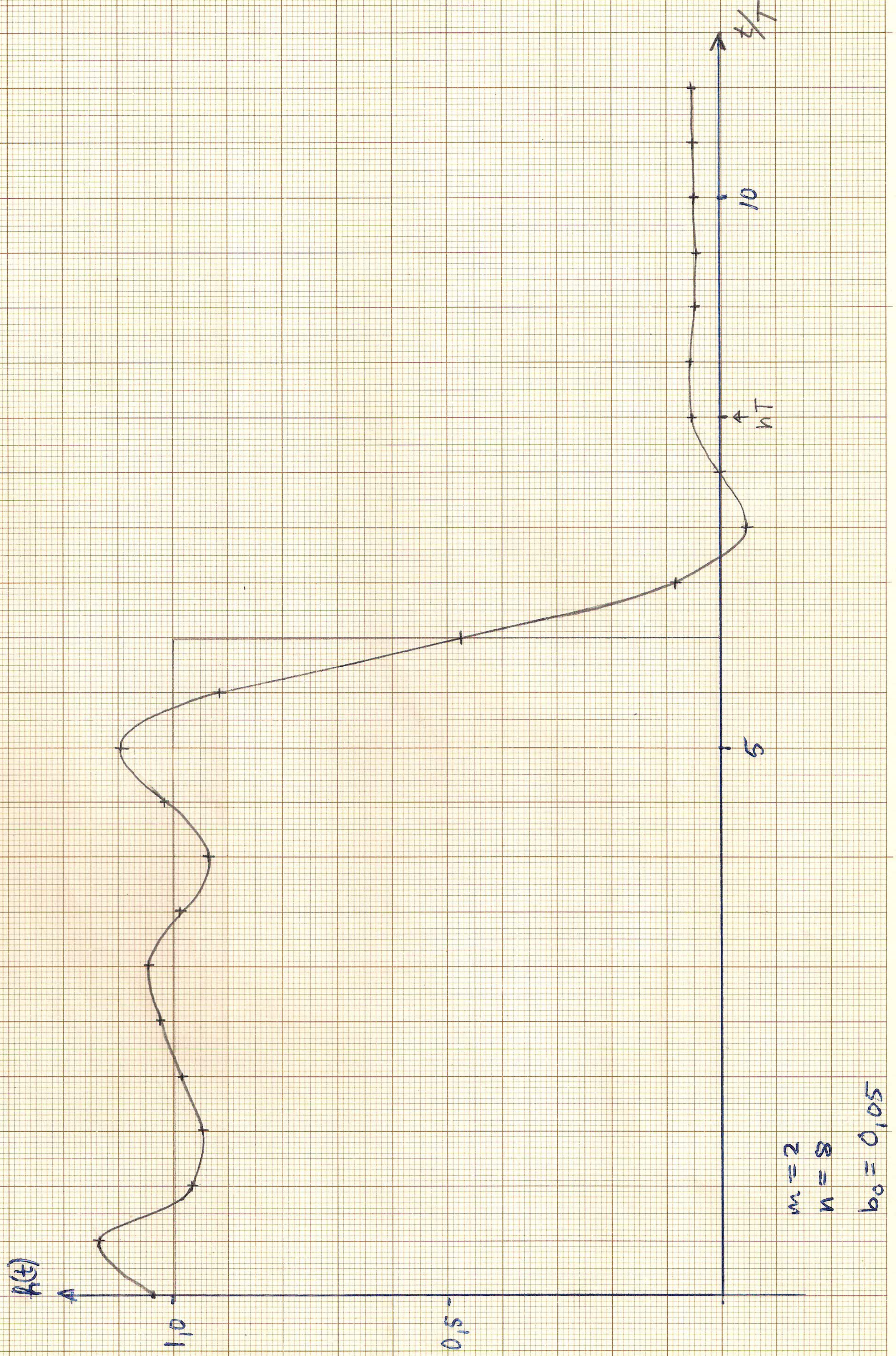
Det färdiga programmet fungerar tillfredsställande. I allmänhet lönar det sig inte att välja m större än 4 till 5. Valet av t=nT har en klar inverkan på felet. För att studera detta närmare kan programmet utökas till att beräkna t.ex. kvadratiska medelfelet. Detta kan beräknas med godtycklig noggrannhet om f(t) är given i procedurform.

Om man inskränker sig till en viss realiseringsmetod, se kap 6, så är det möjligt att låta programmet beräkna de ingående komponentvärdena.

Ex 1.

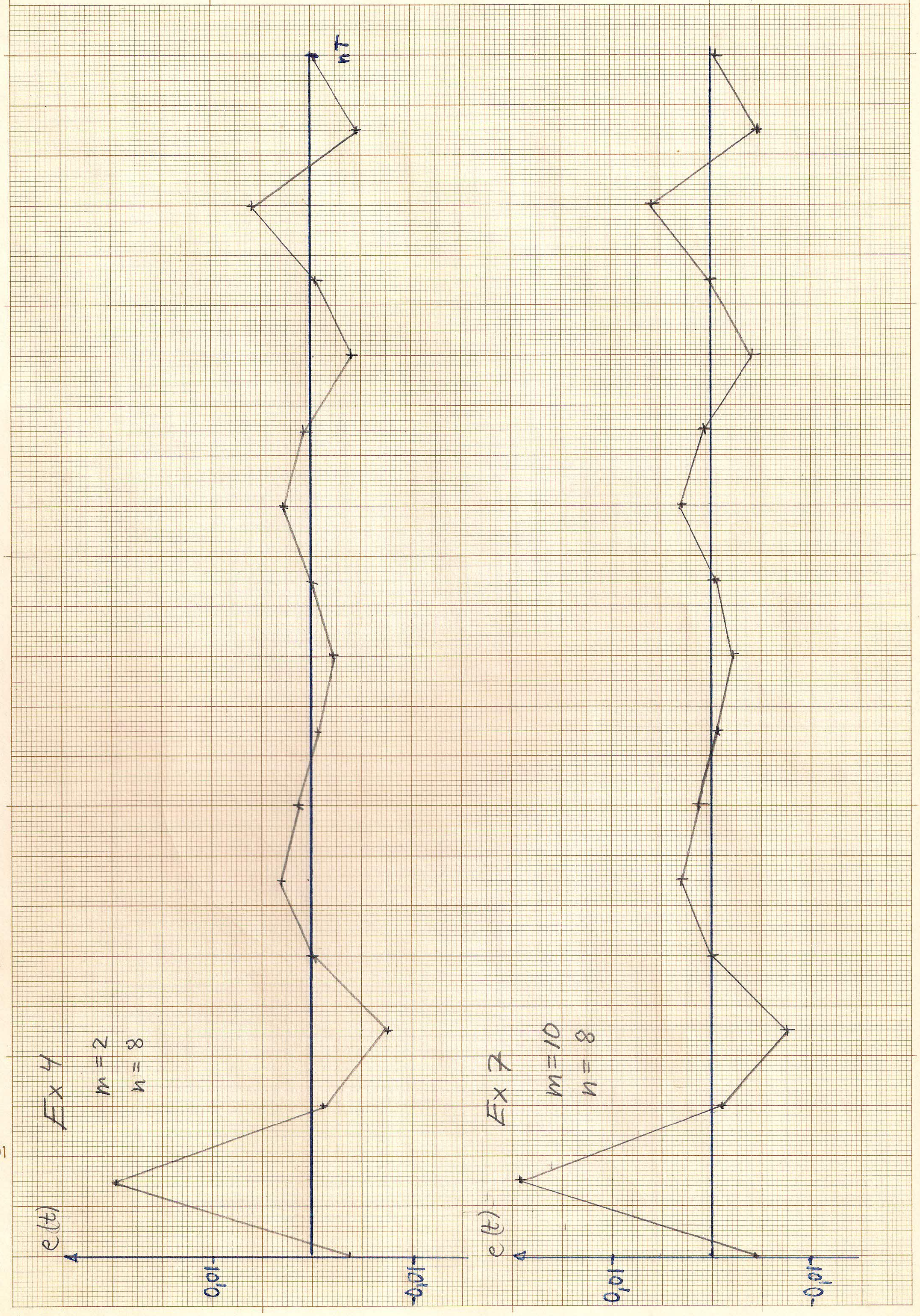


Ex 2.

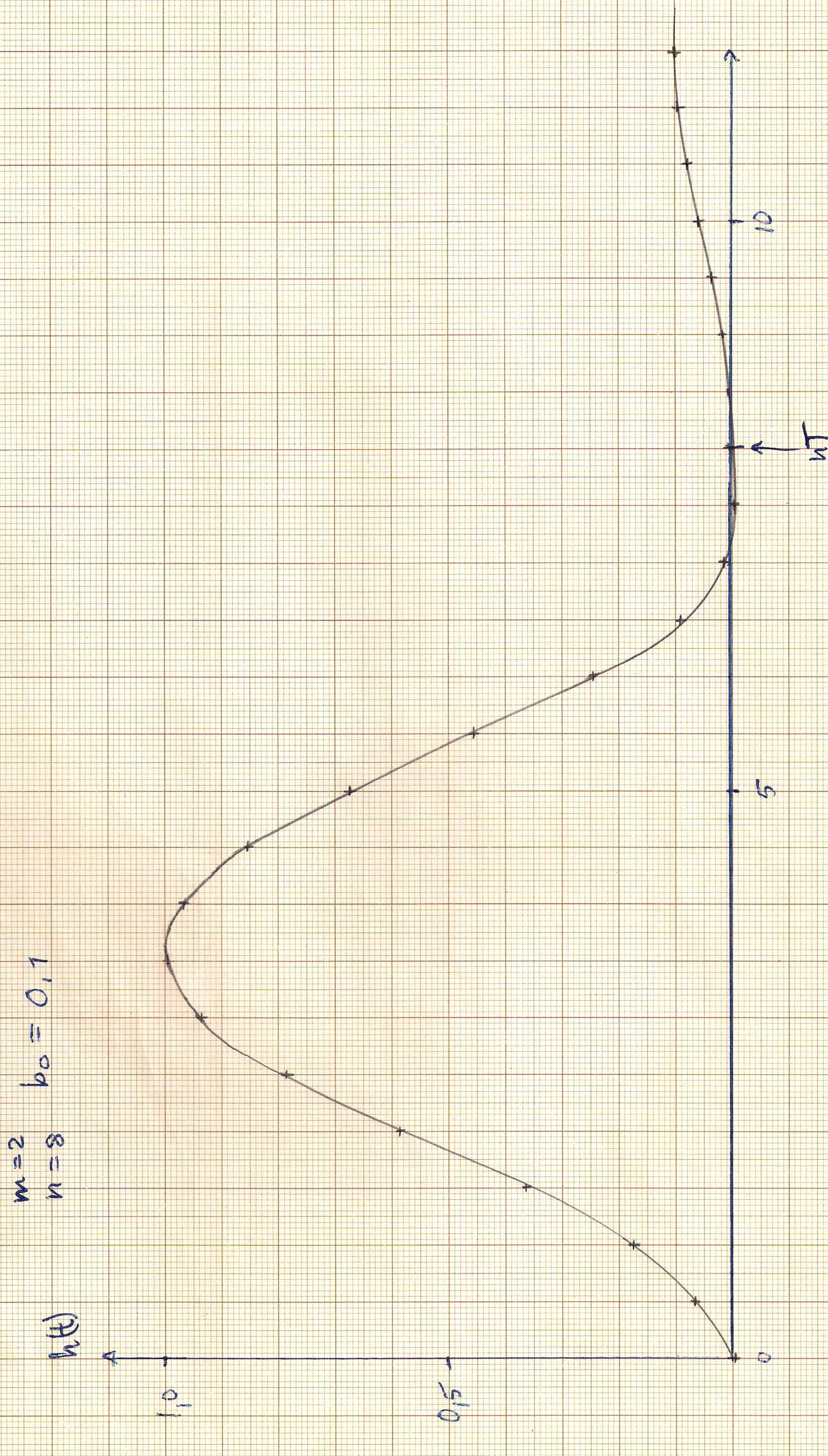


34

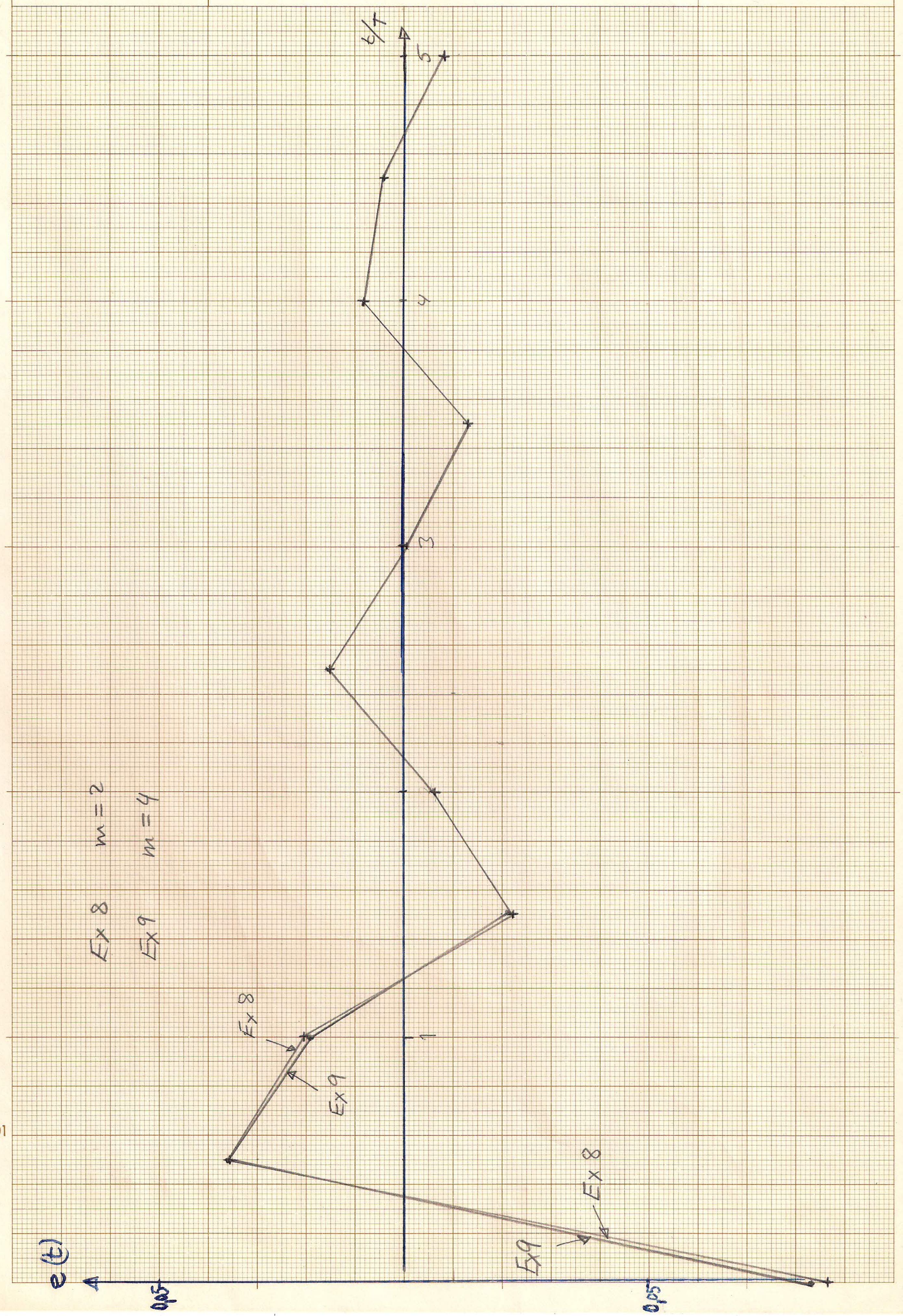
Ex 4 och 7. e(t)



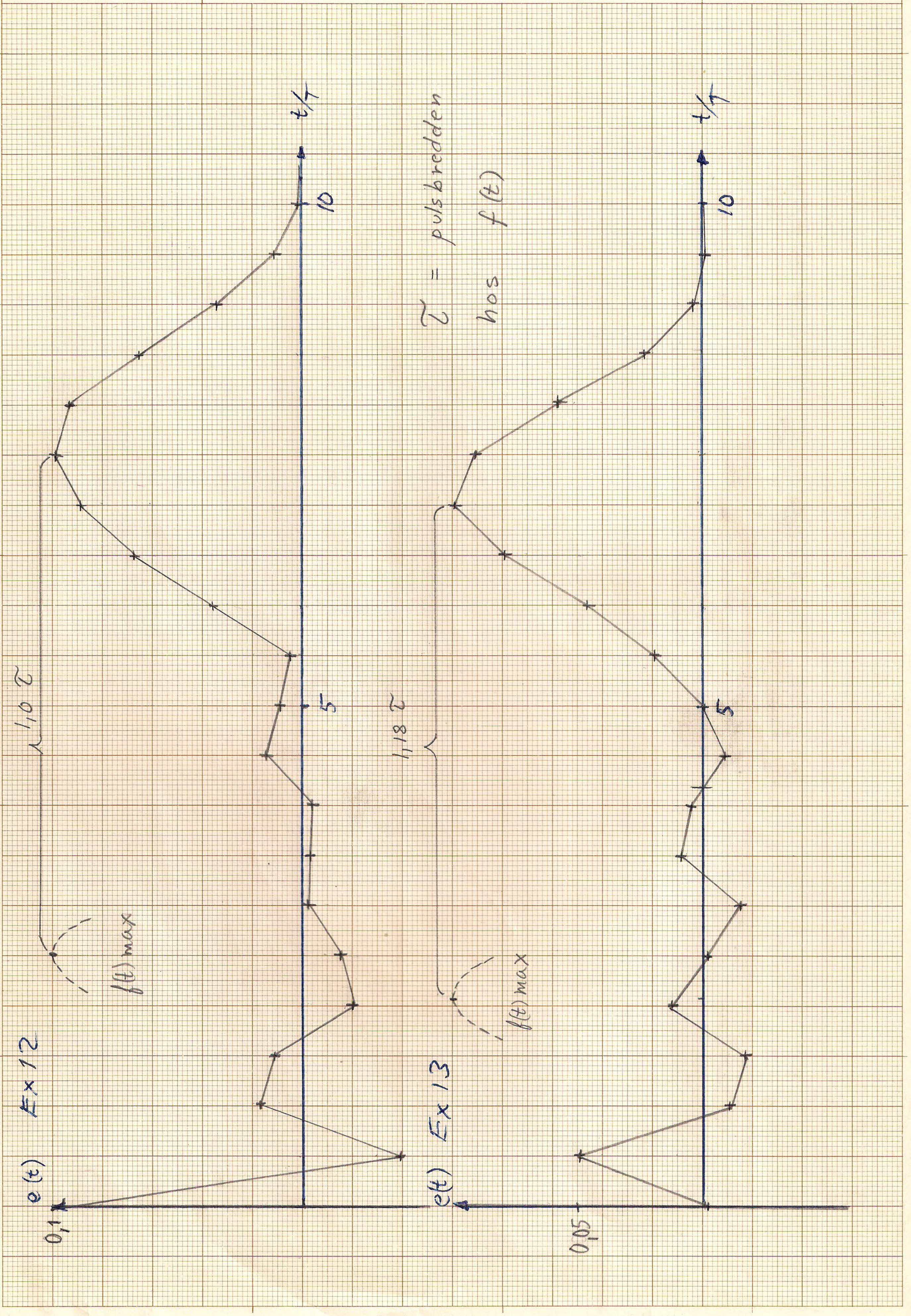
Ex 4



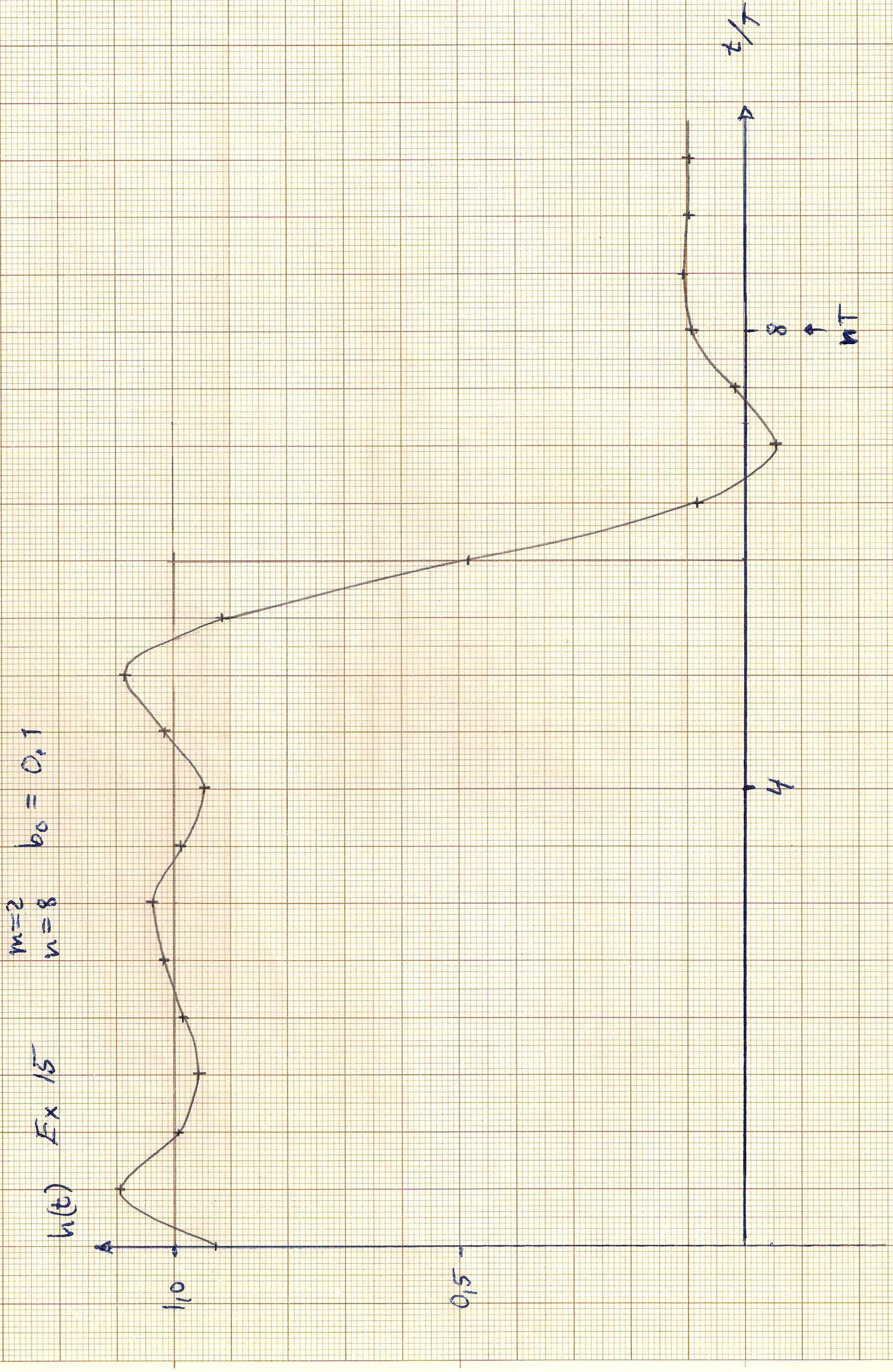
Ex 8 och 9



Ex 12 och 13


t/τ
10
5
τ = pulsbredden
hos f(t)
1,0 τ
1,18 τ
f(t) max
f(t) max
Ex 12
Ex 13
e(t)
e(t)
0,1
0,05

Ex 15.



Utskrift 1

α	M_α	N_α	ω_α	ω_α^2
0	2.14614	0.00000	0.00000	0.00000
1	-1.27287	0.29534	1.25664	1.57914
2	0.19897	0.02523	2.51328	6.31658

b_0	T	m	n
0.10000	1.00000	2	5

b	δ_0
0.63096	-0.46052

h(t)

	0	1
0.00	-0.00082	0.24019
1.00	0.51391	0.78762
2.00	0.92119	0.80477
3.00	0.50607	0.20738
4.00	0.03535	-0.01168
5.00	-0.00008	0.02402
6.00	0.05139	0.07876
7.00	0.09212	0.08048
8.00	0.05061	0.02074
9.00	0.00353	-0.00117
10.00	-0.00001	0.00240

e(t)

-0.00082	-0.00981
0.01391	0.03762
-0.07881	0.05477
0.00607	-0.04262
0.03535	-0.01168

Utskrift 2

ALFA	M	N	W	W↑2
0	5.72529	0.00000	0.00000	0.00000
1	-2.47720	-0.93277	0.78540	0.61685
2	-0.33394	1.48856	1.57080	2.46741
3	0.90410	0.26627	2.35620	5.55168
4	0.16032	-0.51946	3.14160	9.86965

0 1.00000 0.05000 2 8 1.00000

B RESP. SIGMA0 = 0.68766 -0.37447

	0	1
0	1.03576	1.13254
1	0.96878	0.94897
2	0.98249	1.02192
3	1.04497	0.98837
4	0.93512	1.01484
5	1.09685	0.91271
6	0.47325	0.08477
7	-0.04550	0.00012
8	0.05179	0.05663
9	0.04844	0.04745
10	0.04913	0.05110
11	0.05225	0.04942
12	0.04676	0.05074
13	0.05484	0.04564
14	0.02366	0.00424
15	-0.00227	0.00001
16	0.00259	0.00283

0	0.03576	0.13254
1	-0.03122	-0.05103
2	-0.01751	0.02192
3	0.04497	-0.01163
4	-0.06488	0.01484
5	0.09685	-0.08729
6	-0.02675	0.08477
7	-0.04550	0.00012

Utskrift 3

N > 20

a) Utskrift 4

	0	1
0	-0.00374	0.06646
1	0.17740	0.36296
2	0.58668	0.78975
3	0.93431	0.99734
4	0.96775	0.85322
5	0.67386	0.45718
6	0.24601	0.09025
7	0.01333	-0.00459
8	-0.00037	0.00665
9	0.01774	0.03630
10	0.05867	0.07898
11	0.09343	0.09973
12	0.09677	0.08532
13	0.06739	0.04572
14	0.02460	0.00902
15	0.00133	-0.00046
16	-0.00004	0.00066

0	-0.00374	0.01962
1	-0.00121	-0.00763
2	-0.00014	0.00296
3	0.00129	-0.00076
4	-0.00210	-0.00033
5	0.00285	0.00077
6	-0.00399	-0.00017
7	0.00573	-0.00459

ALFA	M	N	W	W↑2
0	2.71924	0.00000	0.00000	0.00000
1	-1.60052	-0.17156	0.78540	0.61685
2	0.15746	0.19067	1.57080	2.46741
3	0.06851	-0.00149	2.35620	5.55168
4	0.02238	-0.01907	3.14160	9.86965

1   1.00000   0.10000   2   8   1.00000

B RESP. SIGMA0 = 0.74989 -0.28782

Utskrift 5

	0	1	2	3
0	-0.00436	0.02906	0.06583	0.11293
1	0.17747	0.26193	0.36316	0.47411
2	0.58665	0.69363	0.78971	0.87097
3	0.93432	0.97710	0.99731	0.99404
4	0.96774	0.92003	0.85329	0.77024
5	0.67387	0.56787	0.45707	0.34751
6	0.24598	0.15881	0.09048	0.04260
7	0.01364	-0.00036	-0.00447	-0.00340
8	-0.00044	0.00291	0.00658	0.01129
9	0.01775	0.02619	0.03632	0.04741
10	0.05867	0.06936	0.07897	0.08710
11	0.09343	0.09771	0.09973	0.09940
12	0.09677	0.09200	0.08533	0.07702
13	0.06739	0.05679	0.04571	0.03475
14	0.02460	0.01588	0.00905	0.00426
15	0.00136	-0.00004	-0.00045	-0.00034
16	-0.00004	0.00029	0.00066	0.00113

0	-0.00436	0.01721	0.01898	0.00961
1	-0.00113	-0.00719	-0.00743	-0.00408
2	-0.00018	0.00229	0.00292	0.00233
3	0.00131	0.00024	-0.00079	-0.00168
4	-0.00211	-0.00166	-0.00026	0.00159
5	0.00286	0.00262	0.00065	-0.00213
6	-0.00402	-0.00339	0.00006	0.00454
7	0.00605	-0.00036	-0.00447	-0.00340

ALFA	M	N	W	W↑2
0	2.71967	0.00000	0.00000	0.00000
1	-1.60024	-0.17194	0.78540	0.61685
2	0.15735	0.19011	1.57080	2.46741
3	0.06794	-0.00191	2.35620	5.55168
4	0.02152	-0.01902	3.14160	9.86965

1 1.00000 0.10000 4 8 1.00000

B RESP. SIGMA0 = 0.74989 -0.28782

c)

	0	1	2	3	4	5
0	-0.00445	0.01776	0.04073	0.06591	0.09565	0.13232
1	0.17750	0.23165	0.29409	0.36313	0.43653	0.51182
2	0.58664	0.65896	0.72709	0.78971	0.84574	0.89422
3	0.93432	0.96527	0.98642	0.99731	0.99774	0.98778
4	0.96774	0.93817	0.89977	0.85329	0.79955	0.73942
5	0.67386	0.60401	0.53120	0.45708	0.38349	0.31246
6	0.24599	0.18594	0.13378	0.09045	0.05628	0.03095
7	0.01358	0.00289	-0.00263	-0.00451	-0.00412	-0.00253
8	-0.00044	0.00178	0.00407	0.00659	0.00957	0.01323
9	0.01775	0.02317	0.02941	0.03631	0.04365	0.05118
10	0.05867	0.06590	0.07271	0.07897	0.08457	0.08942
11	0.09343	0.09653	0.09864	0.09973	0.09977	0.09878
12	0.09677	0.09382	0.08998	0.08533	0.07995	0.07394
13	0.06739	0.06040	0.05312	0.04571	0.03835	0.03124
14	0.02460	0.01859	0.01338	0.00904	0.00563	0.00309
15	0.00136	0.00029	-0.00026	-0.00045	-0.00041	-0.00025
16	-0.00004	0.00018	0.00041	0.00066	0.00096	0.00132

0	-0.00445	0.01248	0.01973	0.01906	0.01340	0.00583
1	-0.00111	-0.00586	-0.00788	-0.00746	-0.00543	-0.00273
2	-0.00018	0.00168	0.00269	0.00292	0.00262	0.00201
3	0.00131	0.00059	-0.00011	-0.00079	-0.00141	-0.00189
4	-0.00211	-0.00193	-0.00129	-0.00026	0.00097	0.00212
5	0.00286	0.00290	0.00214	0.00066	-0.00119	-0.00294
6	-0.00400	-0.00393	-0.00253	0.00003	0.00310	0.00557
7	0.00599	0.00268	-0.00263	-0.00451	-0.00412	-0.00253

ALFA	M	N	W	W↑2
0	2.71955	0.00000	0.00000	0.00000
1	-1.60034	-0.17186	0.78540	0.61685
2	0.15731	0.19022	1.57080	2.46741
3	0.06799	-0.00180	2.35620	5.55168
4	0.02163	-0.01895	3.14160	9.86965

1 1.00000 0.10000 6 8 1.00000

B RESP. SIGMA0 = 0.74989 -0.28782

Utskrift 7

	0	1	2	3	4	5	6	7	8	9
0	-0.00443	0.00882	0.02227	0.03602	0.05040	0.06589	0.08303	0.10235	0.12433	0.14930
1	0.17749	0.20893	0.24352	0.28101	0.32103	0.36314	0.40681	0.45152	0.49673	0.54193
2	0.58664	0.63045	0.67296	0.71387	0.75287	0.78971	0.82418	0.85606	0.88517	0.91131
3	0.93432	0.95403	0.97030	0.98300	0.99202	0.99731	0.99883	0.99658	0.99059	0.98094
4	0.96774	0.95110	0.93118	0.90812	0.88210	0.85329	0.82186	0.78801	0.75191	0.71379
5	0.67386	0.63238	0.58963	0.54593	0.50161	0.45708	0.41273	0.36901	0.32636	0.28521
6	0.24599	0.20909	0.17484	0.14353	0.11536	0.09046	0.06887	0.05054	0.03535	0.02312
7	0.01359	0.00647	0.00144	-0.00185	-0.00373	-0.00450	-0.00447	-0.00387	-0.00290	-0.00172
8	-0.00044	0.00088	0.00223	0.00360	0.00504	0.00659	0.00830	0.01024	0.01243	0.01493
9	0.01775	0.02089	0.02435	0.02810	0.03210	0.03631	0.04068	0.04515	0.04967	0.05419
10	0.05867	0.06305	0.06730	0.07139	0.07529	0.07897	0.08242	0.08561	0.08852	0.09113
11	0.09343	0.09540	0.09703	0.09830	0.09920	0.09973	0.09988	0.09966	0.09906	0.09809
12	0.09677	0.09511	0.09312	0.09081	0.08821	0.08533	0.08219	0.07880	0.07519	0.07138
13	0.06739	0.06324	0.05896	0.05459	0.05016	0.04571	0.04127	0.03690	0.03264	0.02852
14	0.02460	0.02091	0.01748	0.01435	0.01154	0.00905	0.00689	0.00505	0.00354	0.00231
15	0.00136	0.00065	0.00014	-0.00019	-0.00037	-0.00045	-0.00045	-0.00039	-0.00029	-0.00017
16	-0.00004	0.00009	0.00022	0.00036	0.00050	0.00066	0.00083	0.00102	0.00124	0.00149

0	-0.00443	0.00691	0.01467	0.01898	0.02025	0.01904	0.01604	0.01193	0.00735	0.00286
1	-0.00112	-0.00428	-0.00648	-0.00768	-0.00796	-0.00746	-0.00637	-0.00490	-0.00327	-0.00165
2	-0.00018	0.00104	0.00195	0.00255	0.00286	0.00292	0.00279	0.00251	0.00215	0.00174
3	0.00131	0.00088	0.00045	0.00003	-0.00038	-0.00079	-0.00117	-0.00152	-0.00181	-0.00202
4	-0.00211	-0.00205	-0.00184	-0.00145	-0.00092	-0.00026	0.00047	0.00122	0.00192	0.00249
5	0.00286	0.00298	0.00281	0.00235	0.00162	0.00066	-0.00044	-0.00157	-0.00263	-0.00348
6	-0.00401	-0.00412	-0.00376	-0.00291	-0.00161	0.00004	0.00188	0.00369	0.00520	0.00609
7	0.00600	0.00457	0.00144	-0.00185	-0.00373	-0.00450	-0.00447	-0.00387	-0.00290	-0.00172

ALFA	M	N	W	W↑2
0	2.71957	0.00000	0.00000	0.00000
1	-1.60032	-0.17188	0.78540	0.61685
2	0.15732	0.19020	1.57080	2.46741
3	0.06798	-0.00183	2.35620	5.55168
4	0.02161	-0.01897	3.14160	9.86965

1 1.00000 0.10000 10 8 1.00000

B RESP. SIGMA0 = 0.74989 -0.28782

Utskrift 8

	0	1
0	-0.08658	0.15233
1	0.43368	0.72835
2	0.96368	0.98532
3	0.74985	0.40000
4	0.12544	0.00408
5	-0.00866	0.01523
6	0.04337	0.07284
7	0.09637	0.09853
8	0.07498	0.04000
9	0.01254	0.00041
10	-0.00087	0.00152

0	-0.08658	0.03536
1	0.02050	-0.02165
2	-0.00616	0.01548
3	-0.00014	-0.01317
4	0.00846	0.00408

ALFA	M	N	W	W↑2
0	2.71924	0.00000	0.00000	0.00000
1	-1.60122	-0.17236	1.25664	1.57914
2	0.15502	0.19029	2.51328	6.31658

1 0.10000 0.10000 2 5 1.00000

B RESP. SIGMA0 = 0.63096 -0.46052

	0	1	2	3
0	-0.08340	0.02101	0.15368	0.29191
1	0.43307	0.57959	0.72765	0.86296
2	0.96388	1.00912	0.98579	0.89453
3	0.74980	0.57576	0.39953	0.24442
4	0.12517	0.04643	0.00425	-0.01052
5	-0.00834	0.00210	0.01537	0.02919
6	0.04331	0.05796	0.07277	0.08630
7	0.09639	0.10091	0.09858	0.08945
8	0.07498	0.05758	0.03995	0.02444
9	0.01252	0.00464	0.00043	-0.00105
10	-0.00083	0.00021	0.00154	0.00292

0	-0.08340	-0.00914	0.03670	0.04191
1	0.01989	-0.00723	-0.02235	-0.02007
2	-0.00597	0.00912	0.01595	0.01151
3	-0.00020	-0.01106	-0.01364	-0.00558
4	0.00819	0.01628	0.00425	-0.01052

ALFA	M	N	W	W↑2
0	2.71955	0.00000	0.00000	0.00000
1	-1.60037	-0.17191	1.25664	1.57914
2	0.15719	0.19020	2.51328	6.31658

1 0.10000 0.10000 4 5 1.00000

B RESP. SIGMA0 = 0.63096 -0.46052

Utskrift 10

	0	1	2	3	4	5
0	-0.08324	-0.01875	0.06390	0.15375	0.24565	0.33849
1	0.43304	0.53014	0.62923	0.72762	0.82051	0.90155
2	0.96389	1.00129	1.00926	0.98581	0.93188	0.85118
3	0.74980	0.63533	0.51596	0.39951	0.29264	0.20023
4	0.12515	0.06827	0.02869	0.00426	-0.00797	-0.01118
5	-0.00832	-0.00187	0.00639	0.01538	0.02457	0.03385
6	0.04330	0.05302	0.06292	0.07276	0.08205	0.09016
7	0.09639	0.10013	0.10093	0.09858	0.09319	0.08512
8	0.07498	0.06353	0.05160	0.03995	0.02926	0.02002
9	0.01251	0.00683	0.00287	0.00043	-0.00080	-0.00112
10	-0.00083	-0.00019	0.00064	0.00154	0.00246	0.00339

0	-0.08324	-0.03223	0.01071	0.03677	0.04423	0.03652
1	0.01986	0.00107	-0.01418	-0.02238	-0.02261	-0.01619
2	-0.00596	0.00467	0.01264	0.01597	0.01414	0.00807
3	-0.00020	-0.00807	-0.01311	-0.01366	-0.00932	-0.00119
4	0.00818	0.01509	0.01522	0.00426	-0.00797	-0.01118

ALFA	M	N	W	W↑2
0	2.71957	0.00000	0.00000	0.00000
1	-1.60033	-0.17189	1.25664	1.57914
2	0.15730	0.19020	2.51328	6.31658

1 0.10000 0.10000 6 5 1.00000

B RESP. SIGMA0 = 0.63096 -0.46052

49

Utskrift 11

	0	1	2	3	4	5	6	7	8	9
0	-0.08321	-0.04754	-0.00320	0.04656	0.09939	0.15376	0.20879	0.26414	0.31982	0.37603
1	0.43303	0.49098	0.54984	0.60934	0.66889	0.72761	0.78439	0.83788	0.88662	0.92911
2	0.96389	0.98966	1.00534	1.01016	1.00367	0.98582	0.95693	0.91772	0.86920	0.81271
3	0.74980	0.68216	0.61158	0.53983	0.46862	0.39951	0.33388	0.27287	0.21738	0.16802
4	0.12515	0.08886	0.05902	0.03531	0.01726	0.00426	-0.00434	-0.00923	-0.01110	-0.01061
5	-0.00832	-0.00475	-0.00032	0.00466	0.00994	0.01538	0.02088	0.02641	0.03198	0.03760
6	0.04330	0.04910	0.05499	0.06093	0.06689	0.07276	0.07844	0.08379	0.08866	0.09291
7	0.09639	0.09897	0.10053	0.10102	0.10037	0.09858	0.09569	0.09177	0.08692	0.08127
8	0.07498	0.06822	0.06116	0.05398	0.04686	0.03995	0.03339	0.02729	0.02174	0.01680
9	0.01251	0.00889	0.00590	0.00353	0.00173	0.00043	-0.00043	-0.00092	-0.00111	-0.00106
10	-0.00083	-0.00048	-0.00003	0.00047	0.00099	0.00154	0.00209	0.00264	0.00320	0.00376

0	-0.08321	-0.05240	-0.02257	0.00333	0.02342	0.03678	0.04336	0.04374	0.03900	0.03054
1	0.01986	0.00843	-0.00242	-0.01162	-0.01842	-0.02239	-0.02344	-0.02179	-0.01789	-0.01237
2	-0.00596	0.00058	0.00656	0.01138	0.01459	0.01597	0.01546	0.01321	0.00953	0.00488
3	-0.00020	-0.00514	-0.00938	-0.01243	-0.01393	-0.01366	-0.01161	-0.00794	-0.00302	0.00259
4	0.00818	0.01289	0.01580	0.01595	0.01239	0.00426	-0.00434	-0.00923	-0.01110	-0.01061

ALFA	M	N	W	W↑2
0	2.71957	0.00000	0.00000	0.00000
1	-1.60032	-0.17188	1.25664	1.57914
2	0.15732	0.19020	2.51328	6.31658

1 0.10000 0.10000 10 5 1.00000

B RESP. SIGMA0 = 0.63096 -0.46052

	0	1	2	3
0	0.09595	-0.00307	0.05610	0.19634
1	0.36229	0.52281	0.66524	0.78628
2	0.88366	0.95191	0.98249	0.96727
3	0.90285	0.79348	0.65140	0.49442
4	0.34186	0.21014	0.10972	0.04389
5	0.00959	-0.00031	0.00561	0.01963
6	0.03623	0.05228	0.06653	0.07863
7	0.08837	0.09519	0.09825	0.09673
8	0.09029	0.07935	0.06514	0.04944
9	0.03419	0.02101	0.01097	0.00439
10	0.00096	-0.00003	0.00056	0.00196

0	0.09595	-0.05082	-0.03940	-0.02426
1	0.01659	0.02296	0.01124	0.00708
2	-0.02074	-0.00029	-0.01751	0.01507
3	-0.00155	0.01428	-0.00260	-0.00543
4	-0.00384	-0.01046	0.01422	-0.00386

ALFA	M	N	W	W↑2
0	3.45645	0.00000	0.00000	0.00000
1	-1.50547	-0.92534	1.25664	1.57914
2	-0.12681	0.15082	2.51328	6.31658

2 0.10000 0.10000 4 5 1.00000

B RESP. SIGMA0 = 0.63096 -0.46052

i) Utskrift 13

	0	1	2	3
0	-0.00198	0.09239	0.19460	0.31367
1	0.45797	0.62267	0.78695	0.92020
2	0.99291	0.98724	0.90289	0.75678
3	0.57725	0.39554	0.23756	0.11873
4	0.04266	0.00353	-0.00954	-0.00794
5	-0.00020	0.00924	0.01946	0.03137
6	0.04580	0.06227	0.07870	0.09202
7	0.09929	0.09872	0.09029	0.07568
8	0.05772	0.03955	0.02376	0.01187
9	0.00427	0.00035	-0.00095	-0.00079
10	-0.00002	0.00092	0.00195	0.00314

0	-0.00198	0.03509	0.04906	0.01801
1	-0.01105	-0.03133	-0.01729	-0.00332
2	0.01203	0.02548	-0.00151	0.00262
3	-0.01509	-0.01182	-0.00806	0.02323
4	0.00446	0.00353	-0.00954	-0.00794

ALFA	M	N	W	W↑2
0	2.31408	0.00000	0.00000	0.00000
1	-1.42859	0.21980	1.25664	1.57914
2	0.26956	0.03281	2.51328	6.31658

2 0.10000 0.10000 4 5 1.00000

B RESP. SIGMA0 = 0.63096 -0.46052

k)

	0	1	2	3
0	-0.03944	0.14372	0.37407	0.62002
1	0.82673	0.93901	0.92759	0.80215
2	0.60524	0.39236	0.21017	0.08293
3	0.01150	-0.01813	-0.02339	-0.01685
4	-0.00394	0.01437	0.03741	0.06200
5	0.08267	0.09390	0.09276	0.08022
6	0.06052	0.03924	0.02102	0.00829
7	0.00115	-0.00181	-0.00234	-0.00168
8	-0.00039	0.00144	0.00374	0.00620

0	-0.03944	0.06016	0.09092	0.08163
1	0.04753	-0.00124	-0.04851	-0.07095
2	-0.04876	0.00813	0.05212	0.04712
3	0.01150	-0.01813	-0.02339	-0.01685

ALFA	M	N	W	W↑2
0	1.70467	0.00000	0.00000	0.00000
1	-0.97669	0.70272	1.57080	2.46741
2	0.16983	-0.31957	3.14160	9.86965

2 0.10000 0.10000 4 4 1.00000

B RESP. SIGMA0 = 0.56234 -0.57565

Utskrift 15


```
A0              A1              A2  ...
    114.24500       510.41031       621.43990       916.57556       551.98541       413.66964
    132.31733        63.91115         8.61212         3.02303

B0              B1              B2  ...
      8.28958        67.61936       190.05401       273.19139       330.56070       204.59956
    148.24504        45.47197        22.23349         2.87823

ALFA  M               N               W               W↑2
 0        4.08527       0.00000       0.00000       0.00000
 1       -1.59686      -0.31027       0.78540       0.61685
 2       -0.13461       0.96521       1.57080       2.46741
 3        0.56543       0.17284       2.35620       5.55168
 4        0.10380      -0.28600       3.14160       9.86965


  1       0.10000       0.10000       2     8       1.00000

B RESP. SIGMA0 =    0.74989      -0.28782
```

	0	1
0	0.92850	1.09890
1	0.99721	0.95917
2	0.98187	1.01923
3	1.03625	0.98883
4	0.94595	1.01457
5	1.08659	0.91373
6	0.48230	0.08186
7	-0.05418	0.01248
8	0.09285	0.10989
9	0.09972	0.09592
10	0.09819	0.10192
11	0.10363	0.09888
12	0.09460	0.10146
13	0.10866	0.09137
14	0.04823	0.00818
15	-0.00542	0.00125
16	0.00929	0.01099

0	-0.07150	0.09890
1	-0.00279	-0.04083
2	-0.01813	0.01923
3	0.03625	-0.01117
4	-0.05405	0.01457
5	0.08659	-0.08627
6	-0.01770	0.08186
7	-0.05418	0.01248

6. Några metoder att realisera överföringsfunktionen.

6.1. Inledning

Ett allmänt önskemål vid nätrealisering är att de ingående komponenterna avviker så litet som möjligt ifrån den matematiska modell som används vid beräkningarna. Ur denna synpunkt är resistanser och kapacitanser de bästa elementen, medan däremot transformatorer och induktanser mindre väl motsvarar sina enkla matematiska modeller. Induktanserna har en förlustresistans i serie och dessutom en, ofta icke försummbar, strökapacitans.

Genom att tillåta aktiva element i nätet kan man undvara induktanser och transformatorer, dessutom blir syntesen ofta enklare och resulterar i ett bättre nät. De främsta nackdelarna med aktiva element är att de är dyra, och att de fordrar kraftförsörjning.

6.2. Passiva nät

För att en överföringsfunktion skall vara realiserbar med ett ändligt lineärt nät fordras att följande nödvändiga villkor är uppfyllda:

1) $H(s)$ är en rationell funktion av s med reella koefficienter.

2) Nämnaren är ett Hurwitzpolynom.

För passiva nät fordras dessutom:

3) Effektförstärkning mindre än ett för alla frekvenser.

4) $n \leq m+1$, där n är täljarens och m nämnarens gradtal.

Att villkor 1 och 2 är uppfyllda inses lätt. Villkor 3 kan uppfyllas genom att välja en lagom stor konstant faktor framför $H(s)$.

I detta fall gäller förutom villkor 4 att $n < m$.

Konstantresistanslänkar.

Genom att söka poler och nollställen, och därefter uppdela $H(s)$ i lämpliga faktorer, kan man realisera $H(s)$ som en kaskadkoppling av K-R länkar.

En annan väg är att direkt ur $H(s)$ beräkna en korslänk med impedanserna Z_a och Z_b.

Det gäller : $$Z_a = \frac{1}{Z_b} = \frac{1 - H(s)}{1 + H(s)}.$$

Det fordras alltså att Z_a är en PR-funktion, och man kan visa att detta är ekvivalent med följande två villkor:

1) $H(s)$ saknar poler i högra halvplanet.
2) $|H(j\omega)| \leq 1$

Villkor 1 är uppfyllt enligt ovan, se fig 1 sid. 5. Villkor 2 kan uppfyllas genom val av lämplig konstant framför $H(s)$.

Dessa metoder ger som resultat balanserade nät med relativt stort antal komponenter och dessutom omfattande räkningar. De balanserade näten kan undvikas genom användning av aktiva korslänkar, däremot måste man använda ett stort antal induktanser i nätet.

6.3. Aktiva nät.

En godtycklig rationell överföringsfunktion, $H(s)$, kan skrivas som skillnaden mellan två PR-funktioner. I detta fall har vi $H(s)$

given på formen: $H(s) = \sum_{\alpha} \frac{M_\alpha(s-b_0) + N_\alpha \omega_\alpha}{(s-b_0)^2 + \omega_\alpha^2}$.

Betrakta en term ur summan. Den kan skrivas på följande sätt :

$$\frac{k_{1\alpha}(s-b_0)}{s^2 - 2b_0 s + b_0^2 + \omega_\alpha^2} + \frac{k_{2\alpha}\, s}{s^2 - 2b_0 s + b_0^2 + \omega_\alpha^2}$$

där $k_{1\alpha}$ och $k_{2\alpha}$ är positiva eller negativa konstanter. Observera att $b_0 < 0$. De båda termerna är var för sig PR-funktioner, bortsett ifrån eventuellt minustecken. Härav följer att $H(s)$ kan skrivas :

$H(s) = Z_1(s) - Z_2(s)$ där Z_1 och Z_2

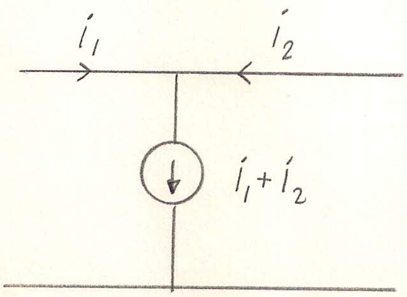
är PR-funktioner.

I fortsättningen betraktas fyra tänkbara metoder för realisering med aktiva nät, varav de två första utgår ifrån att $H(s)$ är given som en kvot mellan två polynom. De två senare metoderna utgår ifrån $H(s)$ som skillnaden mellan två PR-funktioner. I det senare fallet måste man använda sig av induktanser för att realisera $Z_1(s)$ och $Z_2(s)$ separat.

6.3.1. Negativ impedanskonverter.

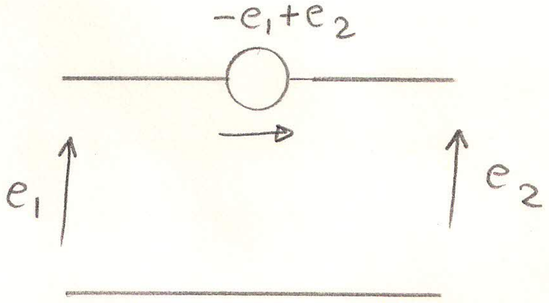
Med hjälp av två RC-nät och en NIC (negativ impedanskonverter) kan man realisera en godtycklig överföringsfunktion, som uppfyller villkoren för aktiva nät.

En NIC är en fyrpol som omvandlar en impedans Z till −Z. Den kan arbeta enligt två olika principer:



$$i_2 = \frac{1}{k}\, i_1$$

Strömінverterande NIC



$$e_2 = -k\, e_1$$

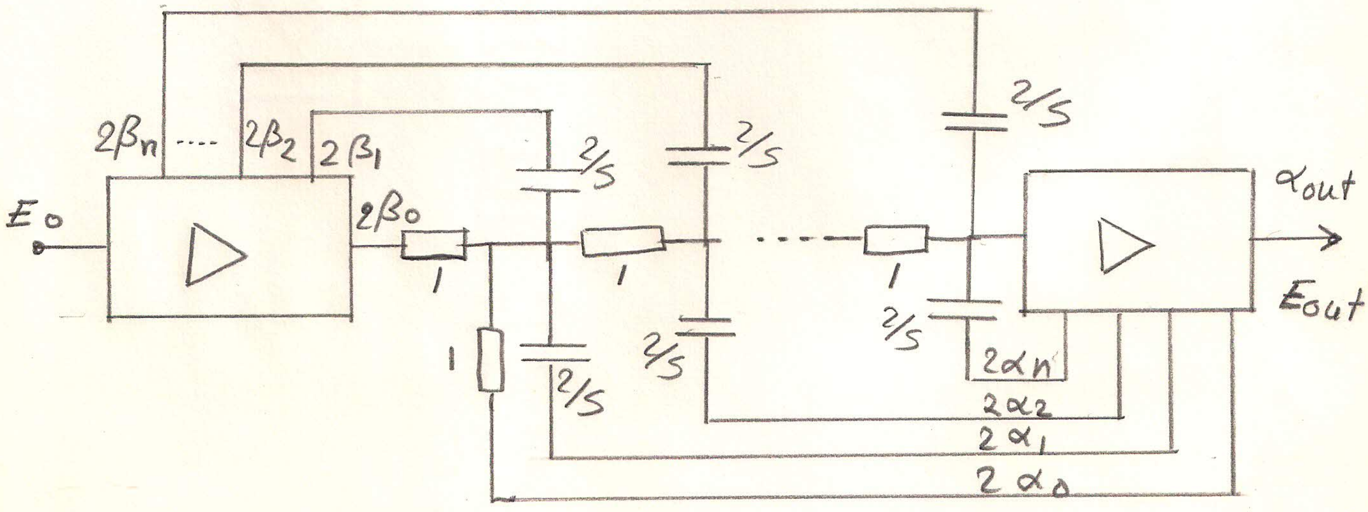
Spänningsinverterande NIC

k är en positiv konstant som kallas konversionsfaktor. Den strömінverterande konvertern är lättast att realisera. För syntesen finns två metoder att välja på varav den ena, Linvill's metod, resulterar i en överföringsimpedans och den andra, Yanagisawas metod, i en spänningsöverföringsfunktion. För beskrivning av metoderna se [2] kap. 6 och [3].

Genom att använda en NIC undviker man alltså induktanser och transformatorer, dessutom blir räkningarna relativt enkla. Vid Yanagisawas metod kan man förenkla ytterligare genom att utgå från att RC-näten utgöres av L-länkar. Linvills metod fordrar att man uppdelar täljaren till H(S) i faktorer. För speciella typer av överföringsfunktioner finns utarbetade metoder för att minimera konversionsfaktorns inverkan på resultatet. Denna varierar nämligen, beroende på yttre förhållanden, hos en verklig NIC.

6-3.2. RC-länkar med återkopplade förstärkare.

Här använder man sig av följande koppling, där $2\alpha_0 \ldots 2\alpha_n$ och $2\beta_0 \ldots 2\beta_n$ betecknar återkopplingsfaktorer.



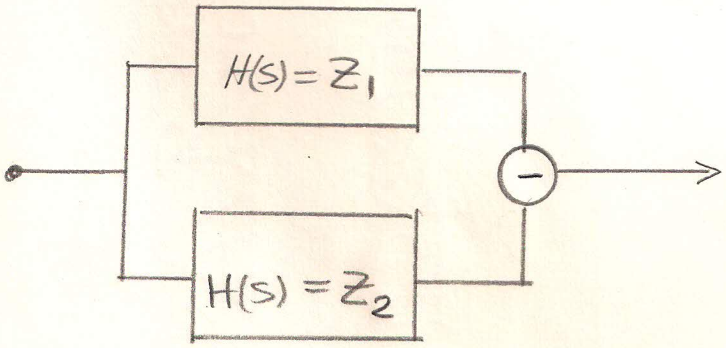
Överföringsfunktionen blir

$$\frac{E_{out}}{E_0} = \alpha_{out} \frac{\beta_0 + \sum_1^n \beta_K Q_K}{1 - \alpha_0 + \sum_1^n (1-\alpha_K) Q_K}$$

där Q_K är ett ortogonalt system av polynom, liknande Chebyshev polynomen. Metoden är användbar för en godtycklig rationell överföringsfunktion $H(s)$ och ger enkla räkningar om $H(s)$ är given med gemensam nämnare. För beskrivning av metoden se [4].

6.3.3. Speciella nät vars överföringsfunktion är en PR-funktion.

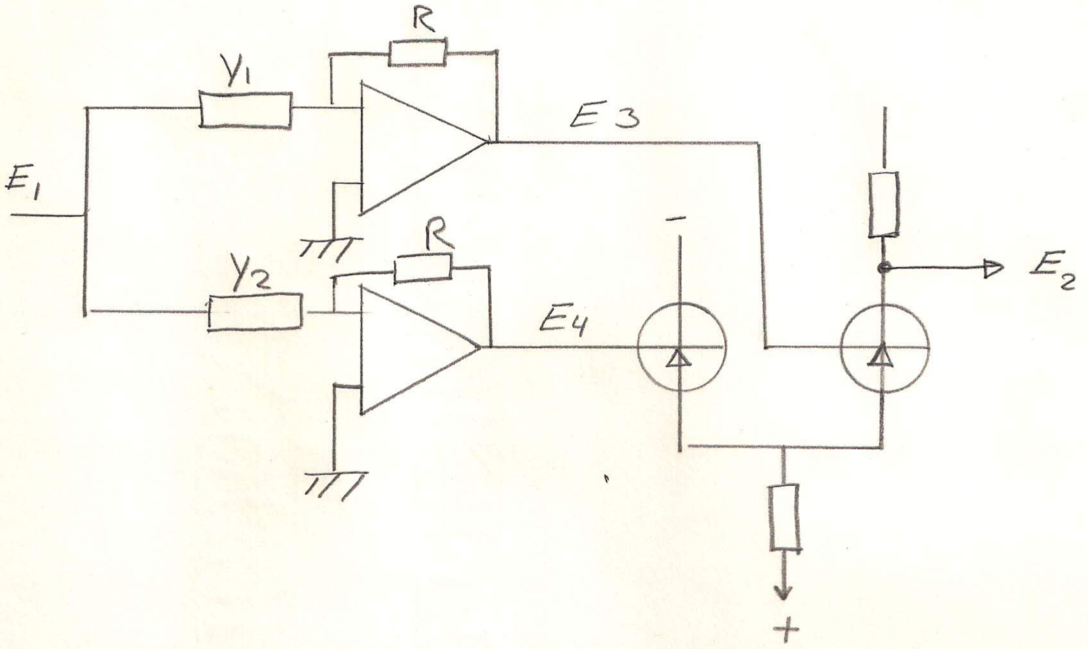
Genom att utnyttja att $H(s) = Z_1 - Z_2$ där Z_1 och Z_2 är PR funktioner så kan man realisera $H(s)$ på följande sätt.



För exempel på nät vars $H(s)$ är PR, se [5] sid 33.

6.3.4. Operationsförstärkare.

Följande koppling ger H(s) som skillnaden mellan addmittanserna Y_1 och Y_2.



$$E_2 = k(E_4 - E_3)$$

$$E_3 = -RY_1E_1$$

$$\frac{E_2}{E_1} = kR(Y_1 - Y_2)$$

$$E_4 = -RY_2E_1$$

6.4. Beräkning av komponentvärden till addmittanserna Y_1 och Y_2 då $H(s) = Y_1 - Y_2$.

Enligt sidan X kan $H(s)$ skrivas på följande sätt, där $k_{1\alpha}$ och $k_{2\alpha}$ är positiva eller negativa konstanter.

$$H(s) = \sum_\alpha \frac{k_{1\alpha}(s-b_0)}{s^2 - 2b_0 s + b_0^2 + \omega_\alpha^2} + \sum_\alpha \frac{k_{2\alpha}\, s}{s^2 - 2b_0 s + b_0^2 + \omega_\alpha^2} =$$

$$= \sum_\alpha \frac{M_\alpha (s-b_0) + N_\alpha \omega_\alpha}{(s-b_0)^2 + \omega_\alpha^2} .$$

Vi söker först $k_{1\alpha}$ och $k_{2\alpha}$.

$$\left.\begin{aligned} M_\alpha &= k_{1\alpha} + k_{2\alpha} \\ -M_\alpha b_0 + N_\alpha \omega_\alpha &= -k_{1\alpha} b_0 \end{aligned}\right\} \text{ varav}$$

$$\begin{cases} k_{1\alpha} = M_\alpha - k_{2\alpha} \\ k_{2\alpha} = \dfrac{N_\alpha \omega_\alpha}{b_0} \end{cases}$$

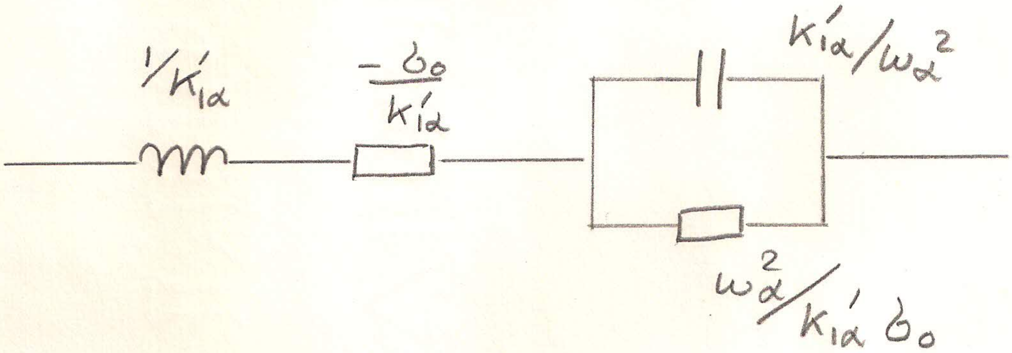
Termer ur $H(s)$ med positiv konstant hänföres till Y_1 och termer med negativ konstant till Y_2. Y_1 och Y_2 kan då bestå av parallellkopplade addmittanser av följande typer:

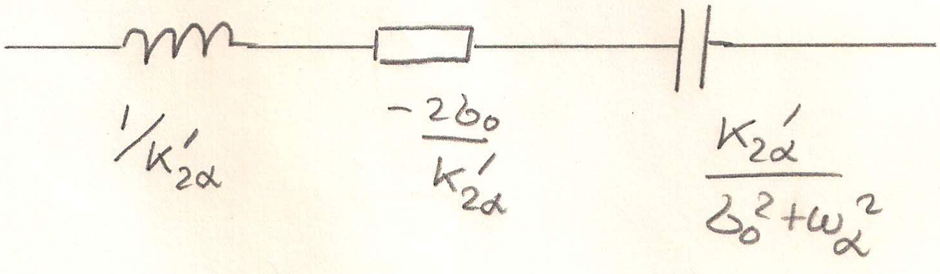
$$\frac{K'_{1\alpha}}{s-b_0+\dfrac{\omega_\alpha^2}{s-b_0}} \quad \text{och} \quad \frac{K'_{2\alpha}}{s-2b_0+\dfrac{b_0^2+\omega_\alpha^2}{s}}$$

där $K'_{1\alpha}$ och $K'_{2\alpha}$ är positiva.
Det kan eventuellt vara nödvändigt att multiplicera $H(s)$ med en konstant, k, för att realiseringsvillkoren skall uppfyllas.
Sätt:

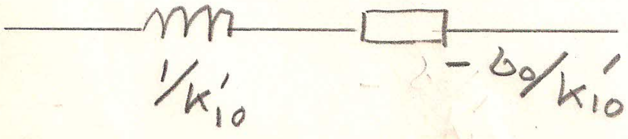
$$K'_{1\alpha} = k/|K_{1\alpha}| \quad \text{och} \quad K'_{2\alpha} = k/|K_{2\alpha}|$$

Deladdmittanserna får följande utseende:





Om $H(s)$ har en pol på reella axeln så får man en addmittans av följande typ:



Observera att elementvärdena är givna som impedanser.