Bab 3. Persamaan Beda dan Operasi Konvolusi

Oleh:
Tri Budi Santoso
Politeknik Elektronika Negeri Surabaya-ITS

Tujuan:
-Siswa mampu membedakan persamaan beda dengan persamaan diferensial
-Siswa mampu menyelesaikan persmasalahan konvolusi diskrit dan menggambarkan contoh aplikasinya

Sub Bab:

3.1.Persamaan Beda Linear Input/Output dengan Koefisien Konstan

3.2. Penyelesaian dengan Rekursi

3.3 Representasi Konvolusi pada Sistem LTI waktu Diskrit

3.4. Studi Kasus Filter Digital

3.1. Persamaan Beda Linear Input/Output dengan Koefisien Konstan

Pertimbangkan sistem waktu diskrit single-input single-output (SISO)

$$y[n]+\sum_{i=1}^{N}a_i\,y[n-i]=\sum_{i=0}^{M}b_i\,y[n-i] \qquad (15)$$

dimana:

n = indek waktu diskrit bernilai integer
x[n] = input
y[n] = output

Asumsi:
- Koefisien-koefisien a_1, a_2, a_3, .. a_N adalah konstanta
- Koefieisn-koefisien b_1, b_2, b_3, .. b_M adalah konstanta.
- N : disebut orde atau dimensi dari sistem.

Persamaan (15) merupakan sistem kausal jika output y[n] pada suatu waktu n hanya tergantung pada nilai-nilai output dan input x[n] di waktu sebelumya

3.2. Penyelesaian dengan Rekursi

Persamaan beda linear input/output dapat diselesaikan dengan prosedur numerik melalui rekursi:

Tulis ulang persamaan (15)

$$y[n] = -\sum_{i=1}^{N} a_i y[n-i] + \sum_{i=1}^{M} b_i x[n-i] \qquad (16)$$

Tetapkan n=0

$y[0]= -a_1y[-1] - a_2y[-2]-.....- a_Ny[-N] + b_0x[0] + b_1x[-1] + ...+b_Mx[-M]$

Sehingga output y[0] pada waktu 0 adalah suatu kombinasi linear pada y[-1], y[-2],.... y[-N] dan x[0], x[-1],...... x[-M].

Tetapkan n=1

$y[1]= -a_1y[0] - a_2y[-1]-.....- a_Ny[-N+1] + b_0x[1] + b_1x[0] + ...+b_Mx[-M+1]$

Sehingga y[1] adalah suatu kombinasi linear pada y[0], y[-1],.... y[-N+1] dan x[1], x[0],...... x[-M+1].

Jika proses ini dilanjutkan, sudah jelas bahwa nilai selanjutnya pada output adalah suatu kombinasi linear pada N nilai terakhir pada output dan M+1 nilai pada input.

Pada setiap step pada komputasi, diperlukan untuk menyimpan hanya N nilai akhir pada output (tentu saja ditambah nilai input). Proses ini disebut sebagai suatu rekursi orde ke-N.

Jika semua a_i adalah nol, persamaan (15) menjadi:

$$y[n] = \sum_{i=0}^{M} b_i x[n-i] \qquad (17)$$

Dalam kasus ini output pada suatu titik waktu tertentu tergantung hanya pada nilai-nilai input x[n], sehingga ouputnya tidak terkomputasi secara recursive. Sistem semacam ini disebut sistem non recursive.

Perlu dicatat bahwa struktur rekursif digambarkan diatas adalah suatu hasil dimensionalitas berhingga. Jika sistem ini dimensionalitasnya tak berhingga, respon output y[n] tidak dapat dihitung secara rekursif

$$y[n] = \sum_{i=0}^{n-1} \frac{1}{n-i} x[i] \qquad n \geq 1$$

adalah infinite dimensional. Dalam kasus ini y[n] tidak dapat diekspresikan dalam bentuk seperti persamaan (15) untuk suatu nilai N, dan selanjutnya y[n] tidak dapat dikomputasi secara rekursif.

Contoh 1

Pertimbangkan sebuah ssstem waktu diskrit yang diberikan persamaan beda input/ouput orde 2 seperti berikut:

$$y[n] - 1{,}5\, y[n-1] + y[n-2] = 2x[n-2] \qquad \text{(i)}$$

Penyelesaian:

Tuliskan kembali persamaan (i) menjadi seperti berikut ini:

$$y[n] = 1{,}5\, y[n-1] - y[n-2] + 2x[n-2] \qquad \text{(ii)}$$

Sekarang tunjukkan bahwa input x[n] adalah fungsi unit-step waktu diskrit u[n] dan nilai ouput awal adalah y[-2] = 2, y[-1] = 1.

Kemudian tetapkan n=0 dalam persamaan (ii) memberikan:

$$\begin{aligned} y[0] &= 1{,}5\, y[-1] - y[-2] + 2x[-2] \\ &= 1{,}5\,(1) - (2) + 2(0) \\ &= -0{,}5 \end{aligned}$$

Dengan n =1 dalam persamaan (ii) memberikan:

$$
\begin{aligned}
y[1] &= 1{,}5\ y[0] - y[-1] + 2x[-1] \\
&= 1{,}5\ (-0{,}5) - (1) + 2(0) \\
&= -1{,}75
\end{aligned}
$$

Lanjutkan proses ini, selajutnya akan memberikan:

$$
\begin{aligned}
y[2] &= 1{,}5\ y[1] - y[0] + 2x[0] \\
&= 1{,}5\ (-1{,}75) - (0{,}5) + 2(1) \\
&= -0{,}125
\end{aligned}
$$

$$
\begin{aligned}
y[3] &= 1{,}5\ y[2] - y[1] + 2x[1] \\
&= 1{,}5\ (-0{,}125) - (-1{,}75) + 2(1) \\
&= 3{,}5625
\end{aligned}
$$

demikian seterusnya.

3. 3. Representasi Konvolusi pada Sistem LTI waktu Diskrit

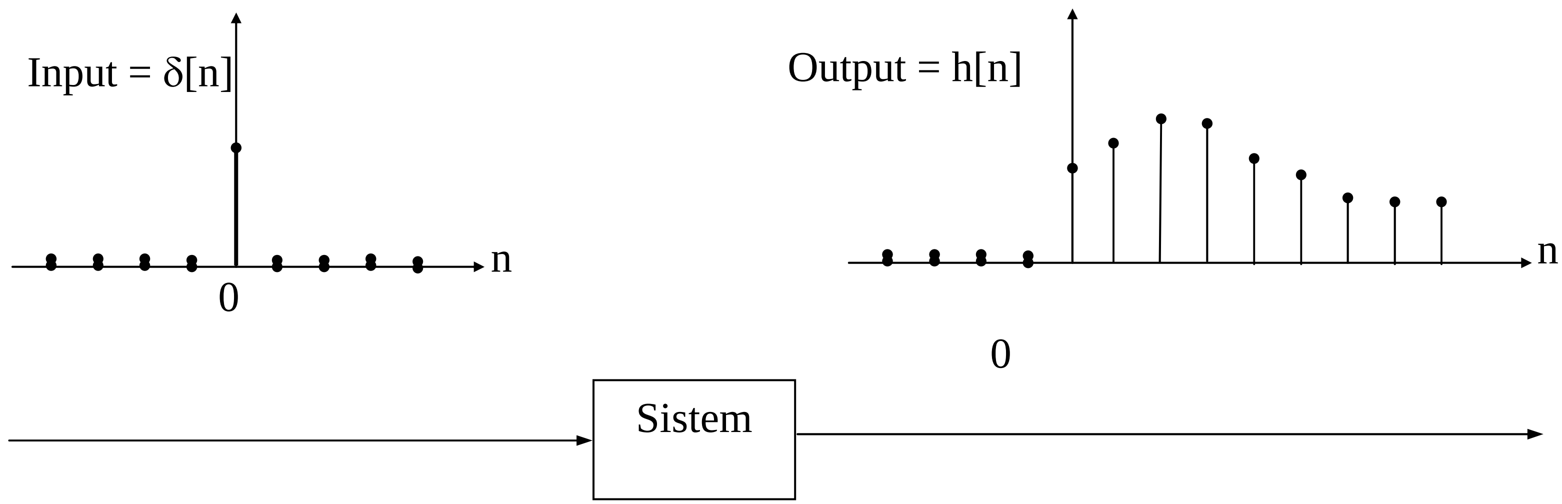
Pertimbangkan sistem waktu diskrit single-input single-output (SISO) dengan input x[n] dan output y[n].
Disini diasumsikan bahwa respon output dihasilkan dari input x[n] tanpa energi awal dalam sistem diprioritaskan untuk aplikasi dari x[n].

Juga diasumsikan bahwa sistem bersifat *kausal*, *linear* dan *time invariant*

Respon Unit Pulsa

Pangil kembali δ[0] = 1 dan δ[n] = 0 untuk semua nilai n bukan nol. Respon h[n] disebut respon unit pulsa pada sistem.

Catatan bahwa dengan δ[n] = -1, -2,… dengan kausalitas respon unit-pulsa h[n] harus nol untuk semua integer n<0



Gambar 3.1. Pembangkitan pada respon unit pulsa

Contoh 2

Pertimbangkan suatu sistem waktu diskrit finite dimensional yang diberikan dengan persamaan beda input/output:

y[n] + ay[n-1] = bx[n]

dimana a dan b adalah konstanta yang. Berikan gambaran respon sistem jika input adalah unit pulsa.

Penyelesaian:

Respon unit pulsa pada sistem ini dapat dikomputasi dengan menyelesaikan persamaan (3-70) dengan kondisi awal y[-1] = 0 dan input x[n]=d[n]. Dari pembicaraan bab 2.3, penyelesaian untuk (3-70) dapat diekspresikan dalam bentuk:

$$y[n] = \sum_{i=0}^{n} (-a)^{n-i} b\delta[i]$$

Konvolusi pada Sinyal Waktu Diskrit

Dua sinyal diskrit x[n] dan v[n], bentuk konvolusi kedua sinyal

$$x[n]*v[n] = \sum_{i=-\infty}^{\infty} x[i]v[n-i]$$

Jika x[n] dan v[n] memiliki nilai 0 untuk semua integer pada n<0, maka:

$$x[n]*v[n] = \begin{cases} 0 & , n = -1, -2, \ldots\ldots \\ \sum_{i=0}^{n} x[i]v[n-i] & , n = 0, 1, 2, \ldots\ldots\ldots \end{cases}$$

Mekanisme Konvolusi Pada Sinyal Diskrit

Contoh penghitung konvolusi pada dua deret nilai integer berikut ini.
Sinyal pertama: x[i]= 1 2 3
Sinyal kedua: v[i]= 2 1 3

Step pertama adalah pembalikan sinyal kedua, v[n] sehingga didapatan kondisi seperti berikut:
Sinyal pertama: x[i] = 1 2 3
Sinyal kedua: v[-i] = 3 1 2

Step ke dua adalah pergeseran dan penjumlahan
```
Sinyal pertama:          1 2 3
Sinyal kedua:        3 1 2
                   ------------------- x
product and sum:     0 0 2 0 0 =2
```

Step ke tiga adalah pergeseran satu step dan penjumlahan
```
Sinyal pertama:           1 2 3
Sinyal kedua:           3 1 2
                   --------------------- x
product and sum:      0 0 1 4 0 =5
```

Step ke empat adalah pergeseran satu step dan penjumlahan
Sinyal pertama: 1 2 3
Sinyal kedua: 3 1 2
-------------------- x
product and sum: 0 0 3 2 6 =11

Step ke lima adalah pergeseran satu step dan penjumlahan
Sinyal pertama: 1 2 3
Sinyal kedua: 3 1 2
-------------------- x
product and sum: 0 0 0 6 3 0 = 9

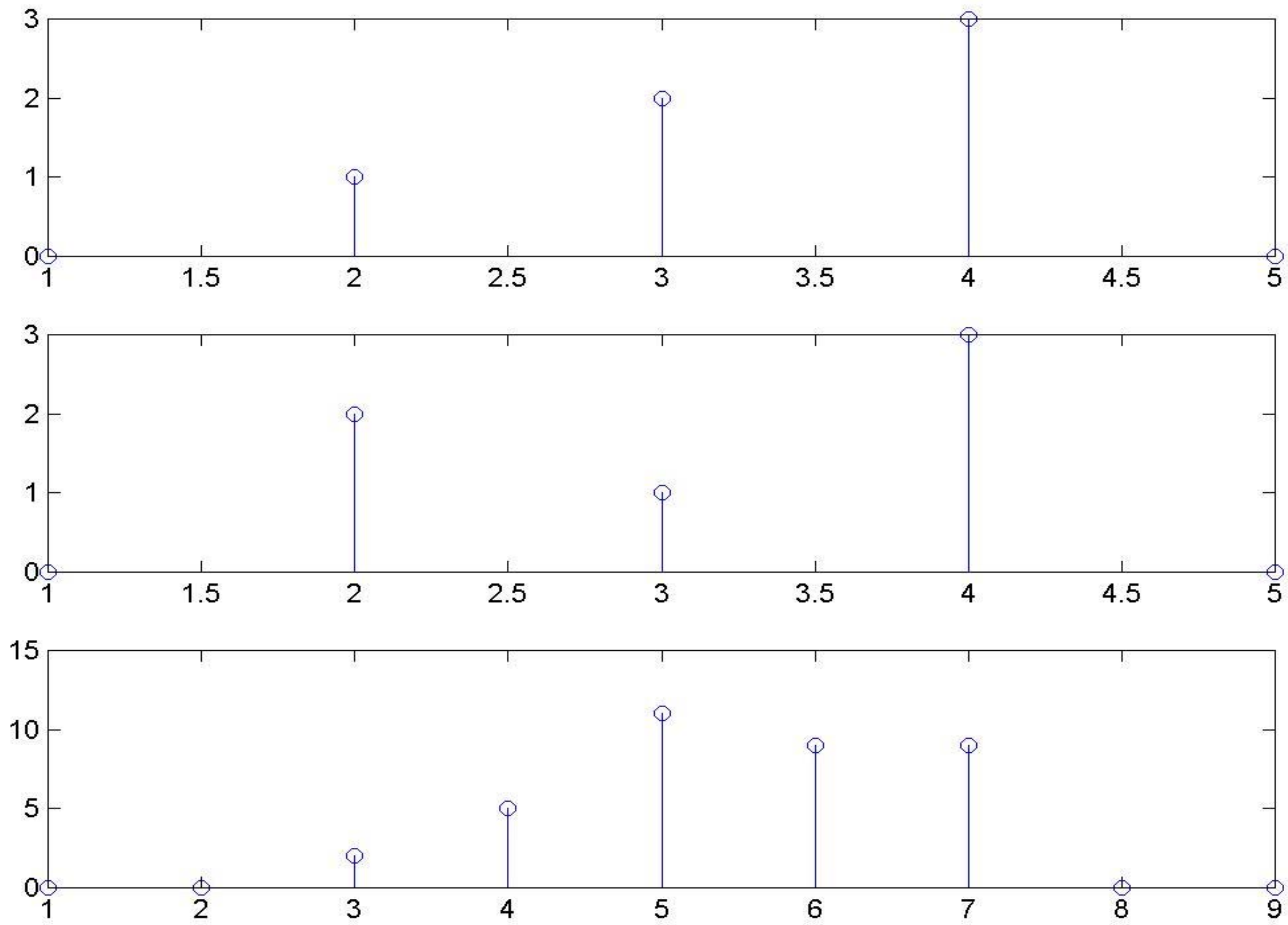
Step ke enam adalah pergeseran satu step dan penjumlahan
Sinyal pertama: 1 2 3
Sinyal kedua: 3 1 2
-------------------- x
product and sum: 0 0 0 0 9 0 0 = 9

Step ke tujuh adalah pergeseran satu step dan penjumlahan

```
Sinyal pertama:                  1 2 3
Sinyal kedua:                        3 1 2
                            ------------------- x
product and sum:              0 0 0 0 0 0 0 = 0
```

Dari hasil product and sum tersebut hasilnya dapat dilihat dalam bentuk deret sebagai berikut: 2 5 11 9 9

```
%File Name: contoh_konvolusi.m
x= [1 2 3] ;
v= [2 1 3];
xv=conv(x,v);
stem(xv)
```



Gambar 3.2. Mekanisme konvolusi

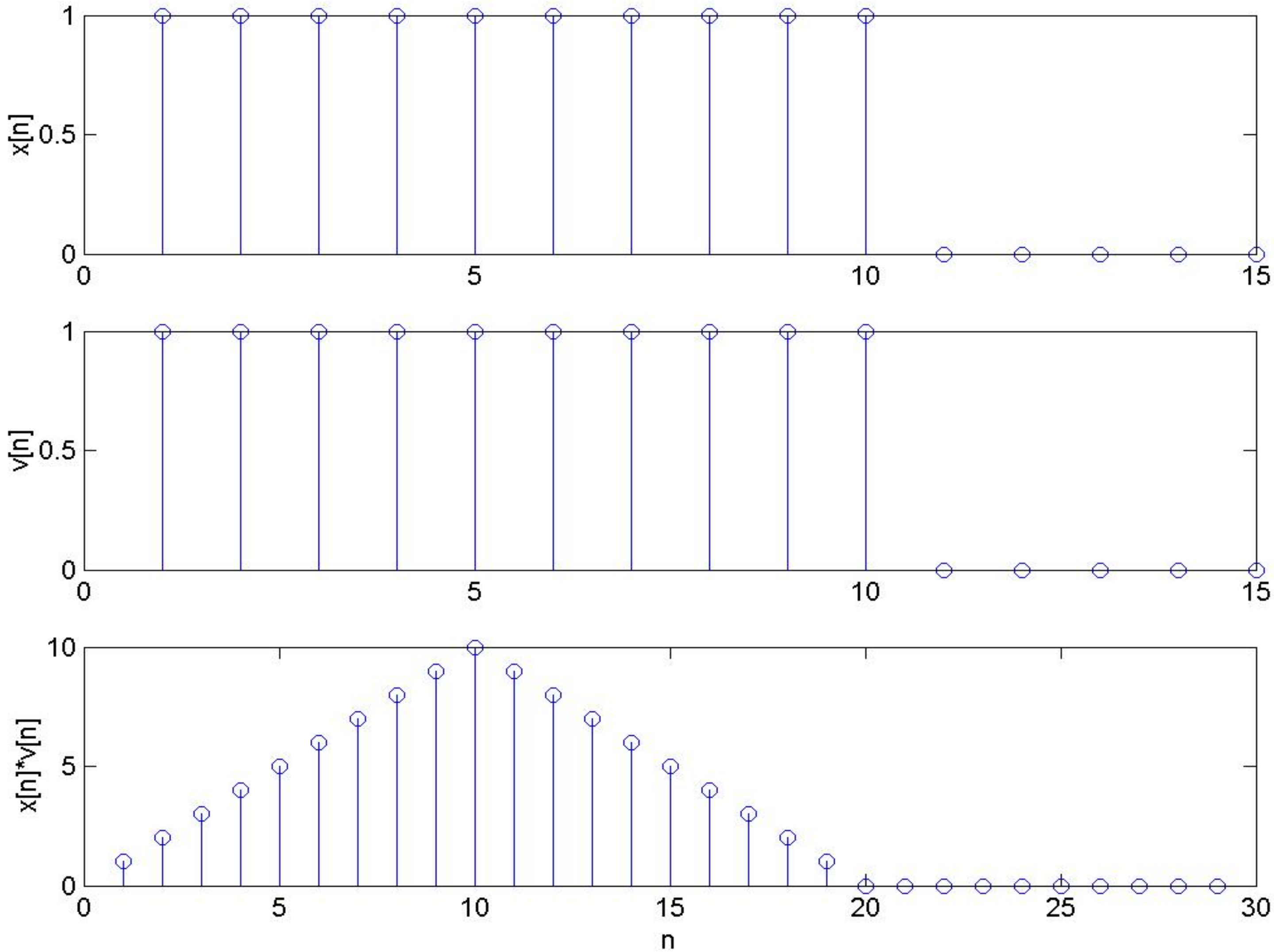
Contoh 3

Berikan gambaran hasil konvolusi dua sinyal persegi x[n] =111111111 dengan v[n]=1111111111.

Penyelesaian:

Dengan langkah yang sama dengan cara yang telah diberikan di atas, kita manfaatkan perangkat lunak Matlab, dan hasilnya seperti Gambar 3.3 berikut ini.

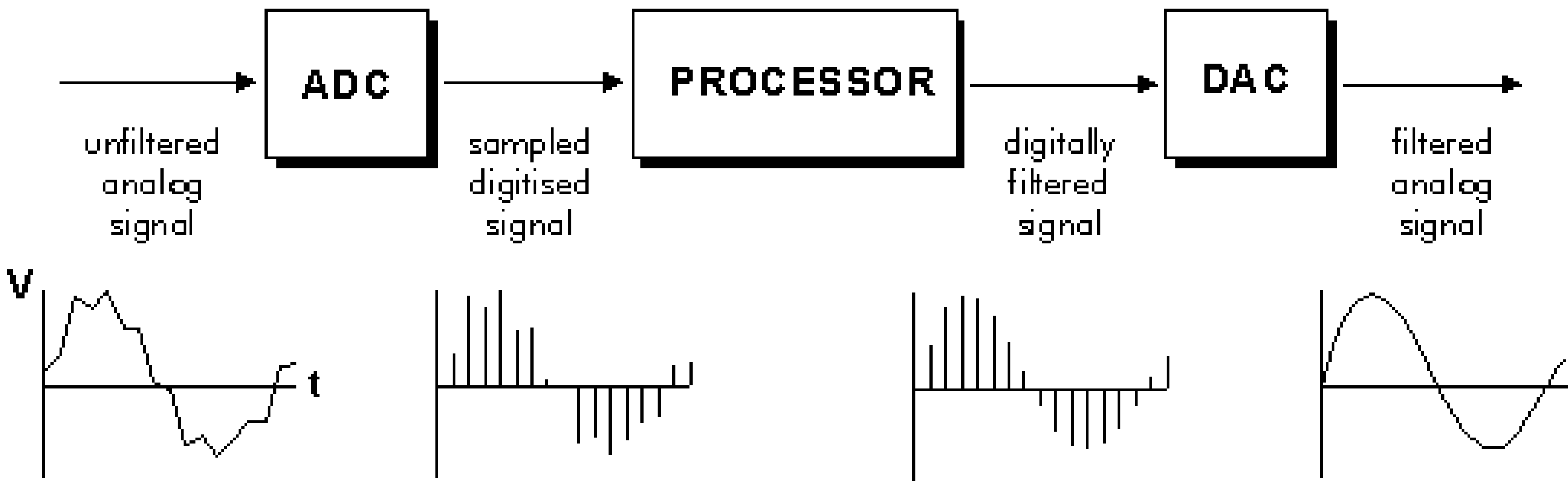
```
%File Name: contoh_konvolusi_2.m
p=[ones(1,10) zeros(1,5)];
x=p;
v=p;
y=conv(x,v);
subplot(3,1,1)
stem(x)
ylabel('x[n]')
subplot(3,1,2)
stem(v)
ylabel('v[n]')
subplot(3,1,3)
stem(y)
xlabel('n')
ylabel('x[n]*v[n]')
```



Gambar 3.3 Hasil konvolusi pada contoh 3

3.4. Studi Kasus Proses Pemfilteran Pada Sinyal Ber-nois

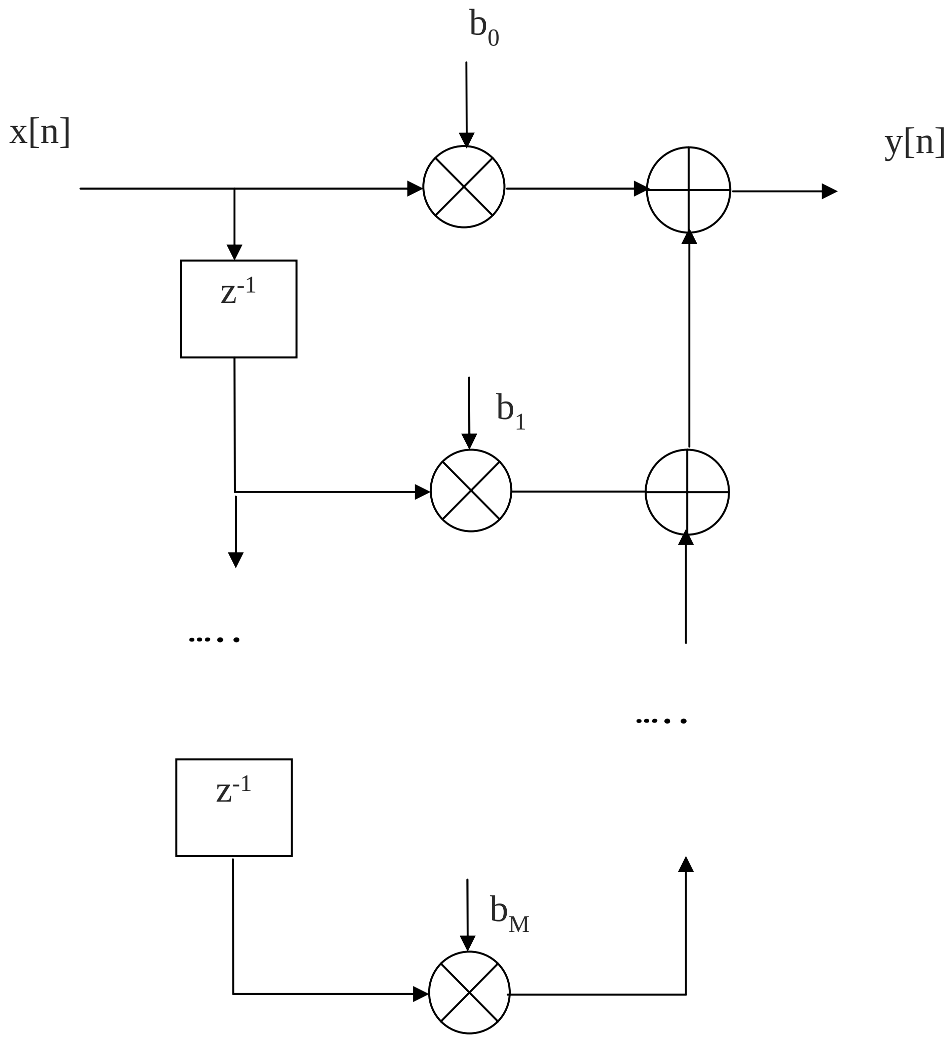
Perhatikan kasus pemfilteran secara digital pada sebuah sinyal ber-noise berikut ini



Kita coba melakukan pemfilteran dengan menggunakan sebuah filter FIR
Sebuah *finite impulse respon filter* (filter FIR) memiliki hubungan input dan output dalam domain waktu diskrit sebagai berikut:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] = \sum_{k=0}^{M} h[k] x[n-k]$$

dimana:
-{*bk*} = koefisien feed forward
- banyaknya (total koefisien) L = M + 1
- M ditetapkan sebagai orde filter FIR



b_0
x[n]
y[n]
z^{-1}
b_1
…. . .
…. . .
z^{-1}
b_M

Pemfilteran

Koefisien-koefisien bn dalam hal ini bernilai:

-0.0031	-0.0050	-0.0068	0.0000	0.0255	0.0731	0.1325
0.1826	0.2023	0.1826	0.1325	0.0731	0.0255	0.0000
-0.0068	-0.0050	-0.0031				

Disini kita gunakan 17 koefisien, atau orde filter = 17

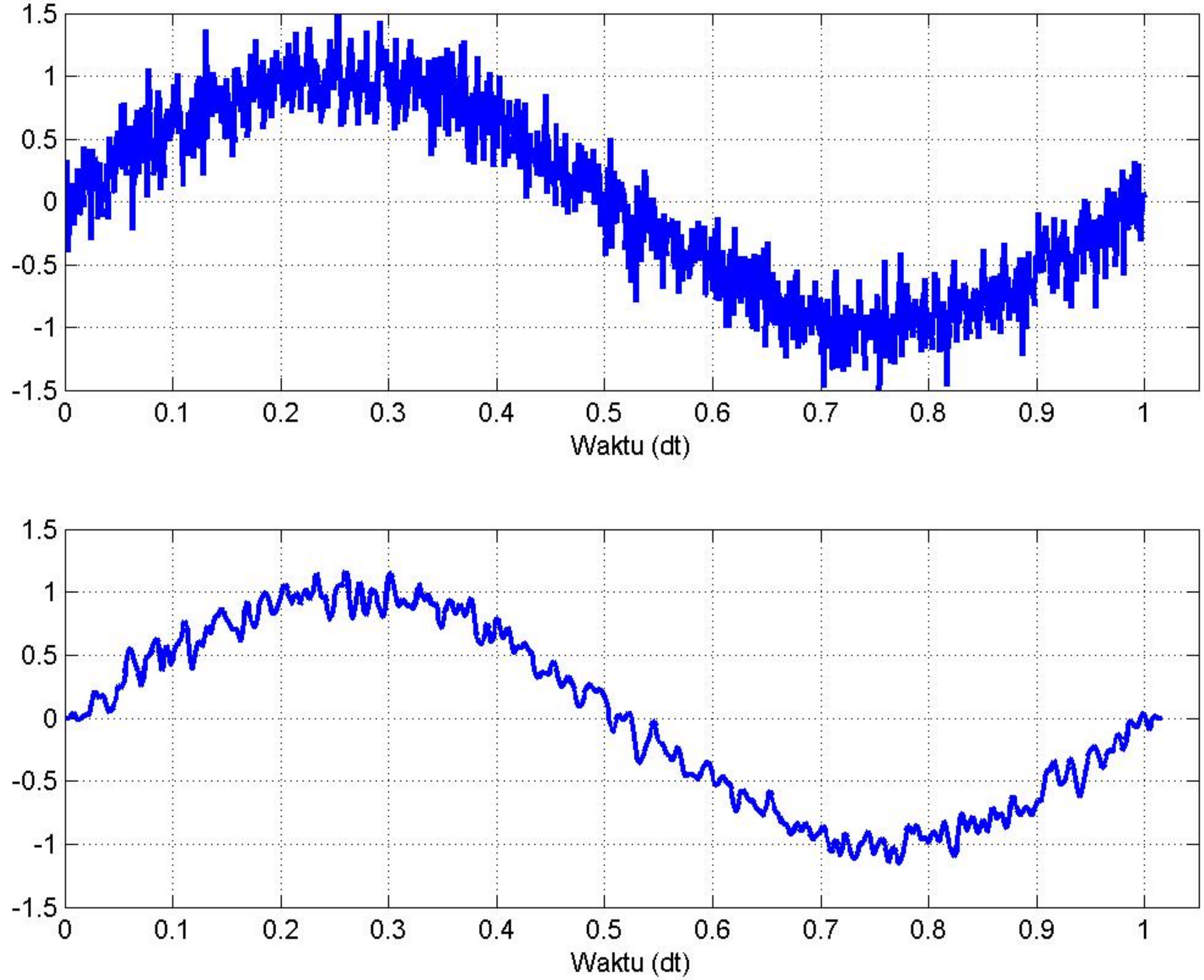
Misalnya input berupa sinyal sinus bernois, maka dengan melakukan operasi konvolusi proses pemfilteran dapat kita lakukan.

- Pada kasus ini kita gunakan simulasi Matlab seperti berikut:

```
%File Name: konvolusi_01.m
clear all;
T=1000;
LPF_01=fir1(16,0.2,'low')
t=1/T:1/T:1;
y=sin(2*pi*t);
tt=length(y);
nois=0.2*randn(1,tt);
y_n = y + nois;
subplot(2,1,1);
plot(t,y_n,'linewidth',2)
axis([0 1.05 -1.5 1.5])
xlabel('Waktu (dt)')
grid on
```

```
%pemfilteran
y_filter=conv(y_n,LPF_01);
subplot(2,1,2);
t_yfil=length(y_filter);
t=1/T:1/T:t_yfil/T;
plot(t,y_filter,'linewidth',2)
axis([0 1.05 -1.5 1.5])
xlabel('Waktu (dt)')
grid on
```

Hasilnya


1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Waktu (dt)
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Waktu (dt)

Penjelasan ….

- Dari gambaran proses pemfilteran di atas dapat dilihat bahwa setelah proses konvolusi sinyal bernois akan mengalami perubahan bentuk menjadi lebih mendekati bentuk sinyal sinus.
- Dengan demikian dapat dikatakan bahwa konvolusi dapat digunakan pada proses pemfilteran