**6 Lọc số và lấy mẫu số tín hiệu**

**6.1 Các bộ loc số lý tưởng**

Một trong những ứng dụng quan trọng của Xử lý số tín hiệu là Lọc số. Các bộ lọc số với ưu điểm của nó đã dần dần thay thế các bộ lọc tương tự.

Có 4 bộ lọc số tiêu biểu là:

* Bộ loc số thông thấp
* Bộ lọc số thông cao.
* Bộ lọc số thông dải
* Bộ lọc số chắn dải

**6.1.1 Bộ lọc thông thấp lý tưởng**

Đáp ứng biên độ được định nghĩa:

|H()| =

(

H(

1

-

**Ví dụ**: cho đáp ưng xung của bộ lọc thông thấp lý tưởng:

|H()| =

Tìm và vẽ đáp ứng xung h(n) trong trường hợp =

**Giải:**

h(n)= =

= .

=

* h(n)= (6.1.1)

thay ta có:

h(n) =



Đồ thi:

syms n;

n=[-8:1:8];

f=0.5\*(sin(n\*pi/2))./(n\*pi/2);

stem(n,f);

ylabel('H(n)');

xlabel('n');

**6.1.2 Bộ lọc thông cao lý tưởng**

Đáp ứng tần số :

|H()| =

(

|H()|

0

***Ví dụ***: Cho đáp ứng xung của bộ lọc số thông cao: (với = 0)

|H()| =

Tìm và vẽ đáp ứng xung h(n) trong trường hợp =

Ta có:

h(n)=

= -

* h(n)= - (6.1.2)

thay ta có:

h(n)= -

***Đồ thị:***

syms n;

n=[-8:1:8];

f=sin(pi\*n)./(pi\*n)-0.5\*(sin(n\*pi/2))./(n\*pi/2);

stem(n,f);

ylabel('H(n)');

xlabel('n');



***Nhận xét:***

* Giống với bộ lọc thông thấp lý tưởng với bộ lọc thông cao lý tưởng đáp ứng xung h(n) đối xứng tại mẫu n=0 vì là tuyến tính và =0
* Trong công thức (6.1.2) chính là đáp ứng xung của bộ lọc thông tất (All- pass filter) với đáp ứng xung :

|H()| =1

**6.1.3 Bộ lọc số thông giải lý tưởng**

Đáp ứng xung :

|H()| =

(

-

***Nhận xét*** :

* Bộ lọc thông dải lý tưởng có đáp ứng xung h(n) đối xứng tại mẫu n=0 vì là tuyến tính và =0
* Các tham số bao gồm:

:tần số cắt dưới

:tần số cắt trên

:dải thông

:dải chắn

:dải chắn

***Ví dụ***: Cho đáp ứng xung của bộ lọc số thông cao: (với = 0)

|H()| =

Tìm và vẽ đáp ứng xung h(n) trong trường hợp =.

**Giải**:

Ta có:

h(n)=

= - (6.1.3)

Thay =. Vào biểu thức (6.1.3)

h(n)= -

Đồ thị:

{ n=-5:1:5;

f=(sin(n\*pi/2)-(sin(n\*pi/3)))./(n\*pi);

stem(n,f);

ylabel('H(n)');

xlabel('n');



Nhận xét:

* Nếu 2 bộ lọc thông thấp có tần số cắt và thì ta có bộ lọc thông dải chính là hiệu của 2 bộ lọc thông thấp đã nói trên
* Khi ta nhận được bộ lọc thông dải hẹp thường được sử dụng làm bộ lọc cộng hưởng.

**\*KẾT LUẬN VỀ BỘ LỌC SỐ LÝ TƯỞNG:** Các bộ lọc số lý tưởng không thể thực hiện về mặt vật lý vì đáp ứng xung h(n) có độ dài vô cùng và h(n) không nhân quả.

**6.1.4 Bộ lọc số chắn dải lý tưởng**

Đáp ứng xung:

|H()| =

(

Đồ thị đáp ứng xung:

H()|

**Ví dụ:** Cho đáp ứng xung của bộ lọc số chắn dải lý tưởng

|H()| =

Tìm và vẽ đáp ứng xung h(n) trong trường hợp =.

**Giải:**

Ta có:

h(n)=

= -

= -[ - ] (6.1.4)

Thay = vào (6.1.4)

h(n)=- [ - ]

Đồ thị:

n=-7:1:7;

f=sin(pi\*n)./(pi\*n)-(sin(n\*pi/2)-(sin(n\*pi/3)))./(n\*pi);

stem(n,f);

ylabel('H(n)');

xlabel('n');



**6.2 Các bộ lọc số thực tế**

Các bộ lọc số thực tế được đặc trưng bởi các thông số kỹ thuật trong miền thời gian lien tục với 4 tham số chính là:

* : Bộ gợn song dải thông
* : Bộ gợn sóng dải chắn
* Tần số giới hạn dải thông
* :Tần số giới hạn giải chắn

Ngoài ra còn có tham số phụ là: =:-

Các tham số kỹ thuật của thông cao, thông dải và chắn dải có thể tự suy ra .

**6.3 Bộ vi phân**

**\* Định nghĩa:** Hệ thông dùng để vi phân tín hiệu được gọi là bộ vi phân

Đáp ứng tần số của bộ vi phân lý tường:

H()=j

Từ đó ta có đáp ứng xung của bộ vi phân lý tưởng:

h(n)=IFT [H()]=

=

* h(n)=

Nhận xét: đáp ứng xung của bộ vi phân là phản đối xứng , tức là:

h(n)=-h(n)

h(0)= 0

Đồ thị đáp ứng xung:

{ n=-7:1:7;

f=cos(pi\*n)./n;

stem(n,f);

ylabel('H(n)');

xlabel('n');}



**6.4** **Bộ lọc số HILBERT**

Đáp ứng xung của bộ lọc số HILBERT được định nghĩa như sau:

|H()| =

Thực tế bộ lọc số Hilbert chính là bộ lọc thông tất (All –pass filter) làm nhiệm vụ dip ha 1 góc 900

Bộ lọc số Hilbert thường được sử dụng thường xuyên trong các hệ thống thông tin và xử lý số tín hiệu

Biến đổi H() dưới dạng đáp ứng biên độ và đáp ứng pha ta có:

H()= |H()|.

|H()| = 1 -

=

Đáp ứng xung của bộ lọc số Hilbert:

h(n)=IFT [H()]=

=-

* h(n)= (6.4.1)

Từ biểu thức (6.4) ta thất rằng h(n) có chiều dài vô hạn và không nhân quả. Ngoài ra h(n) phản đối xứng tức là:

h(n)=-h(-n)

h(0)=0 (6.4.2)

Đồ thị đáp ứng xung:

{ n=-7:1:7;

f=pi/2\*(sin(n\*pi/2)).^2./n;

stem(n,f);

ylabel('H(n)');

xlabel('n');}



**6.5 Lấy mẫu tín hiệu**

**6.5.1 Đinh lý lấy mẫu**

Giả sử chúng ta có một tín hiệu tương tự mang năng lượng hữu hạn x(t), tức là tín hiệu đó đã thõa mãn điều kiện :

<

Cũng có thể nói rằng tín hiệu trên có bề rộng phổ là hữu hạn:

Ở đây là mốc tần số hữu hạn.

Bởi vì hữu hạn nên ta có thể viết dưới dạng chuổi Fourier như sau :

()= (6.5.1.1)

Với :

= (6.5.1.2)

Theo định lý Fourier thì :

(t)= (6.5.1.3)

Nhưng do phổ () hữu hạn nên ta có thể viết:

(t)= (6.5.1.4)

**Chương 7: Tổng hợp các bộ lọc số**

**7.1 Bộ lọc FIR pha tuyến tính**

**7.1.1 Đáp ứng tần số của pha ( Đáp ứng pha)**

* Cái lợi cơ bản nhất của bộ lọc FIR khi tính toán h(n) là khả năng tính toán theo bộ lọc pha tuyến tính. Tức là chúng ta có thể gia công bộ lọc FIR bằng cách coi đáp ứng tần số H() của nó có pha tuyến tính. Cũng vậy, tín hiệu qua dải thông của bộ lọc sẽ xuất hiện chính xác ở đầu ra với độ trễ đã cho, bởi vì chúng ta đã biết chính xác đáp ứng pha của nó.
* Giả sử h(n) là đáp ứng xng của bộ lọc FIR xác định với các mẫu n=0;1;…;N-1 tức là
* L[h(n)]=[0;N-1]=N
* Hàm truyền đạt
* Đáp ứng tần số
* H() =
* hoặc
* Nếu h(n) là thực thì theo biến đổi Fourier đối với tín hiệu rời rạc ta có:
  + = -
    - hoặc
* Vậy có thể nói:
  + : là hàm chẵn (đối xứng)
  + là hàm lẻ (phản đối xứng)
* Do tuần hoàn với chu kì 2π vậy chỉ xét và trong khoảng 0≤⍵≤2π và trong trường hợp đặc biệt nếu h(n) là thực thì là hàm chẵn và là hàm lẻ trong khoảng một chu kỳ, vì vậy ta chỉ cần nghiên cứu trong khoảng 0≤⍵≤π

Nhưng ở chương 3 ta thấy rằng khi cho các chỉ tiêu kỹ thuật của bộ lọc thực tế ( thì cho theo , nhưng cách biểu diễn pha φ(⍵) lại bất tiện vì có thể lấy giá trị âm hoặc dương nhưng bao giờ cũng lấy giá trị dương. Vì vậy để đảm bảo thuận lợi cho việc thiết kế bộ lọc FIR pha tuyến tính chúng ta sẽ dùng cách biểu diễn dưới dạng A() và pha θ(⍵).

)

Cách biểu diễn pha θ(⍵) sẽ cho ta đơn giản hóa phương pháp nghiên cứu pha.

Dưới đây chúng ta sẽ xét chi tiết các đặc điểm của bộ lọc FIR pha θ(⍵) tuyến tính.

**7.1.2 BỘ LỌC SỐ FIR PHA TUYẾN TÍNH**

**a) Điều kiện pha tuyến tính**

H(

Θ(⍵) = β - α⍵ -π≤⍵≤π

* Thời gian lan truyền tín hiệu τ

(5.3.2.1)

Trong trường hợp này τ = -α

Xét 2 trường hợp

* Trường hợp 1: θ(⍵) = -α⍵ -π≤⍵≤π

[cosα⍵ - jsinα⍵]

Ngoài ra có thể tính theo FT[h(n)]

Vậy ta có

→

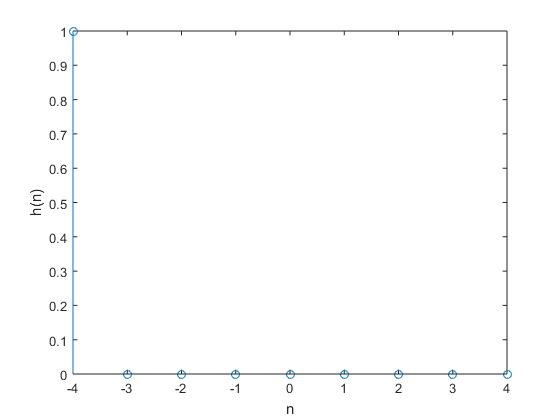
Vì sin0 = 0; cos0 = 1 ta có:

Đến đây lại xét 2 trường hợp α = 0 và α ≠ 0

* Nếu α = 0 →

→ với n và giá trị h(0) ≠ 0

Chọn tùy ý

**Hình 5.3.2.1 sẽ cho ta một ví dụ về cách chọn h(n)

Hình 5.3.2.1.

*n=-4:1:4;*

*N=length(n);*

*delta=[1 zeros(1,N-1)];*

*stem(n,delta);*

*xlabel('n');*

*ylabel('h(n)');*

Đây là trường hợp h(n) tầm thường, không cho chúng ta hiệu quả gì cả, vậy ta bỏ qua không xét nữa.

* Nếu α ≠ 0 →

sin =

Vậy ta có

Phương trình trên có dạng của 1 chuỗi Fourier, chúng ta có thể tìm thấy một nghiệm của nó và nghiệmnày là duy nhất. Nghiệm có dạng như sau

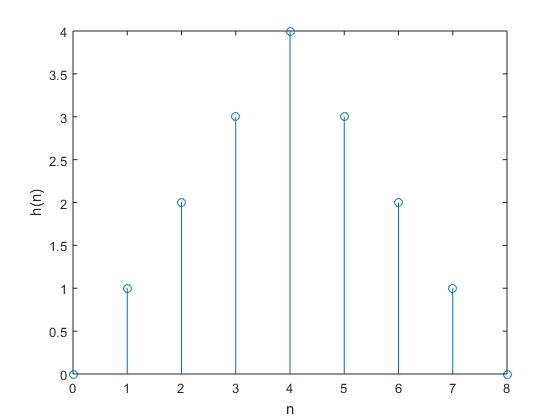
α = ;

h(n) = h(N – 1 – n ) (0≤n≤N – 1 )

**Ví dụ 5.3.2.1:**

*Cho bộ lọc số FIR pha tuyến tính θ(ω) = -αω có N = 7, h(0) = 1, h(1) = 2, h(2) = 3, h(3) = 4. Hãy tìm α và vẽ h(n).*

***Giải***

Vậy .

*x1 = [ 0 1 2 3];*

*x2 = [ 3 2 1 0];*

*n = 0:size(x,2)-1;*

*x = [ x1 4 x2 ];*

*xlim([-1 7]);*

*ylim([0 5]);*

*stem(n,x);*

*xlabel('n');*

*ylabel('h(n)')*

*stem(n,x);*

Tâm đối xứng nằm tại

Hình 5.3.2.2

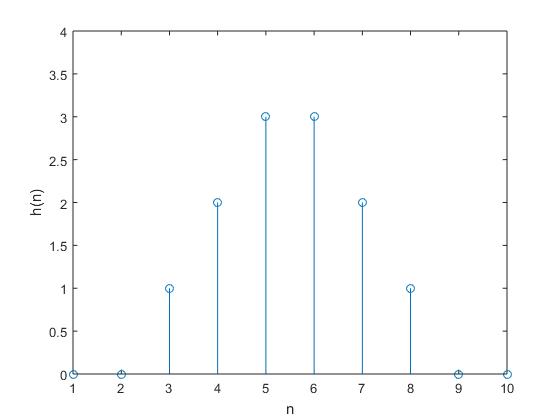
**Ví dụ 5.3.2.2 :**

*Cho bộ lọc số FIR pha tuyến tính θ(⍵) = -α(⍵) có N = 6, h(0) = 1, h(1) = 2,*

*h(2) = 3. Hãy tìm ⍵ và vẽ h(n).*

***Giải***

Vậy

Hình 5.3.2.3

*x1 = [ 0 0 1 2 3];*

*x2 = [ 3 2 1 0 0];*

*x = [ x1 x2 ];*

*stem(x);*

*xlabel('n');*

*ylabel('h(n)')*

*xlim([1 10]);*

*ylim([0 4]);*

Tâm đối xứng nằm giữa 2 điểm

Nhận xét:

* Đối với giá trị α này, đáp ứng xung h(n) là đối xứng.
* Nếu N lẻ thì α là một số nguyên và tâm đối xứng của đáp ứng xung trùng với mẫu
* Nếu N chẵn thì α là một số không nguyên và tâm đối xứng của đáp ứng xung nằm giữa

Đặc điểm quan trọng nhất đối với bộ lọc số FIR pha tuyến tính này kà tính đối xứng của đáp ứng xung h(n) mà sau này chúng ta sẽ có rất nhiều ứng dụng quan trọng.

* Trường hợp 2

Chứng minh tương tự như trường hợp 1 ta có:

Và phương trình có nghiệm duy nhất như sau:

h(n) = - h(N – 1 – n)

**Ví dụ 5.3.2.3:**

*Cho bộ lọc số FIR pha tuyến tính θ(⍵) = β -α(⍵) có N = 7, h(0) = 1, h(1) = 2,*

*h(2) = 3. Hãy tìm ⍵ và vẽ h(n).*

***Giải:***

=3

Vậy

Đồ thị Hình 5.3.2.4

**Ví dụ 5.3.2.4:**

*Cho bộ lọc số FIR pha tuyến tính θ(⍵) = β -α(⍵) có N = 6, h(0) = 1, h(1) = 2,*

*h(2) = 3. Hãy tìm ⍵ và vẽ h(n).*

***Giải:***

=2,5

Vậy

Đồ thị Hình 5.3.2.5

*x1=[0 01 2 3];*

*x2=[-3 -2 -1 0 0];*

*x=[x1 0 x2];*

*stem(x);*

*xlim([0 10]);*

*ylim([-pi pi]);*

*xlabel('n');*

*ylabel('h(n)');*

*figure*

*x1=[0 0 1 2 3];*

*x2=[-3 -2 -1 0 0];*

*x=[x1 x2];*

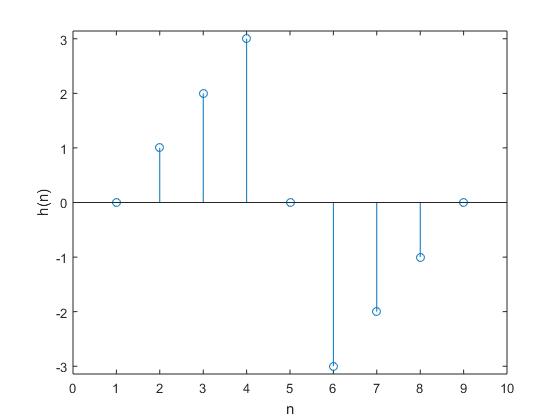
*stem(x);*

*xlim([1 10]);*

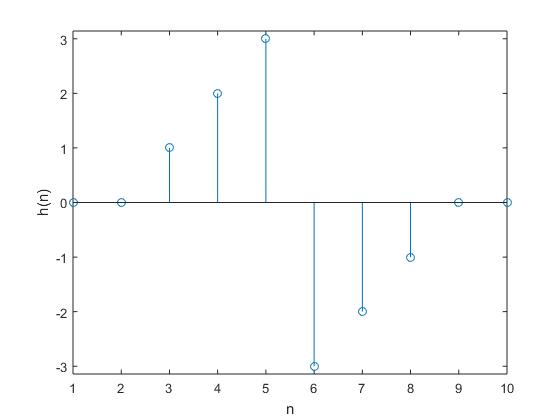
*ylim([-pi pi]);*

*xlabel('n');*

*ylabel('h(n)');*

****

Hình 5.3.2.4



Hình 5.3.2.5

Nhận xét:

* Đối với một giá trị của N, chỉ có một giá trị α đảm bảo pha tuyến tính
* Đối với giá trị α này, đáp ứng xung h(n) là phản đối xứng
* Nếu N lẻ thì α là số nguyên và tâm đối xứng của h(n) trùng với mẫu , và h() = 0
* Nếu N chẵn thì α là số không nguyên và tâm đối xứng của h(n) nằm giữa hai mẫu , và
* Đặc điểm quan trọng nhất ở đây đối với bộ lọc FIR pha tuyến tính

θ(⍵) = β -α(⍵) là h(n) phả đối xứng

b) Tổng kết

* Từ kết quả ở trên đối với bộ lọc số FIR pha tuyến tính chúng ta cha ra làm 4 loại bộ lọc.
* Bộ lọc loại 1: h(n) đối xứng, N lẻ
* Bộ lọc loại 2: h(n) đối xứng, N chẵn
* Bộ lọc loại 3: h(n) phản đối xứng, N lẻ
* Bộ lọc loại 4: h(n) phản đối xứng, N chẵn

Cả 4 loại bộ lọc số FIR pha tuyến tính ở trên cho phép xác định đáp ứng tần số sao cho thỏa mãn các chỉ tiêu kỹ thuật của bộ lọc.

**7.1.3 Đáp ứng tần số của các bộ lọc FIR pha tuyến tính**

**7.1.3.1 Trường hợp đáp ứng xung đối xứng, N lẻ (Bộ lọc FIR loại 1)**

* Ta có:

Áp dụng tính đối xứng của h(n), ta chia tổng này ra làm 3 phần:

Đổi biến số ở thành phần thứ 3: n = N – 1 – m

Ta có

Sau đó áp dụng h(n) = h(N – 1 – n) và biến đổi tiếp ta thu được kết quả:

ở đây: a(0) = h

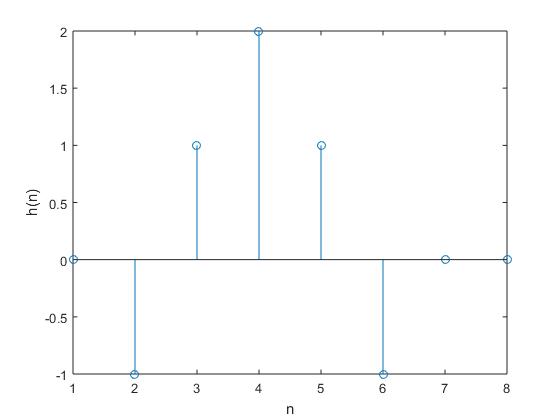
a(n) = 2h (1 ≤ n ≤)

So sánh với biểu thức:

Ta có

**Ví dụ 5.4.1.1:**

*Cho đáp ứng xung h(n) của bộ lọc FIR pha tuyến tính như trên hình 5.4.1.1.*



*x= [0 -1 1 2 1 -1 0 0];*

*xlim([-1 2]);*

*ylim([-1 2]);*

*stem(x)*

*xlabel('n');*

*ylabel('h(n)');*

Hình 5.4.1.1

*Hãy tìm a(n), và .*

***Giải:***

N = 5, N – 1 = 4

Vậy

Đồ thị của a(n) cho trên hình 5.4.1.2.

*x= [0 2 2 -2 0];*

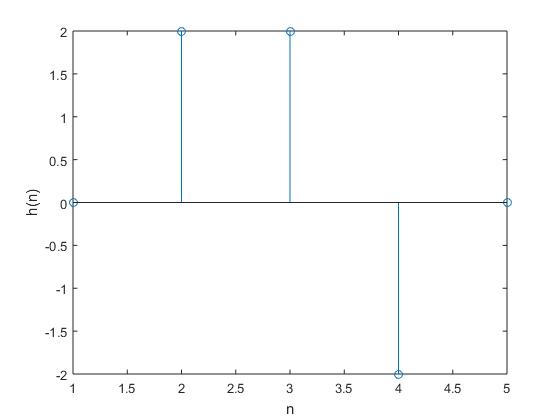
*xlim([-1 3]);*

*ylim([-2 2]);*

*stem(x)*

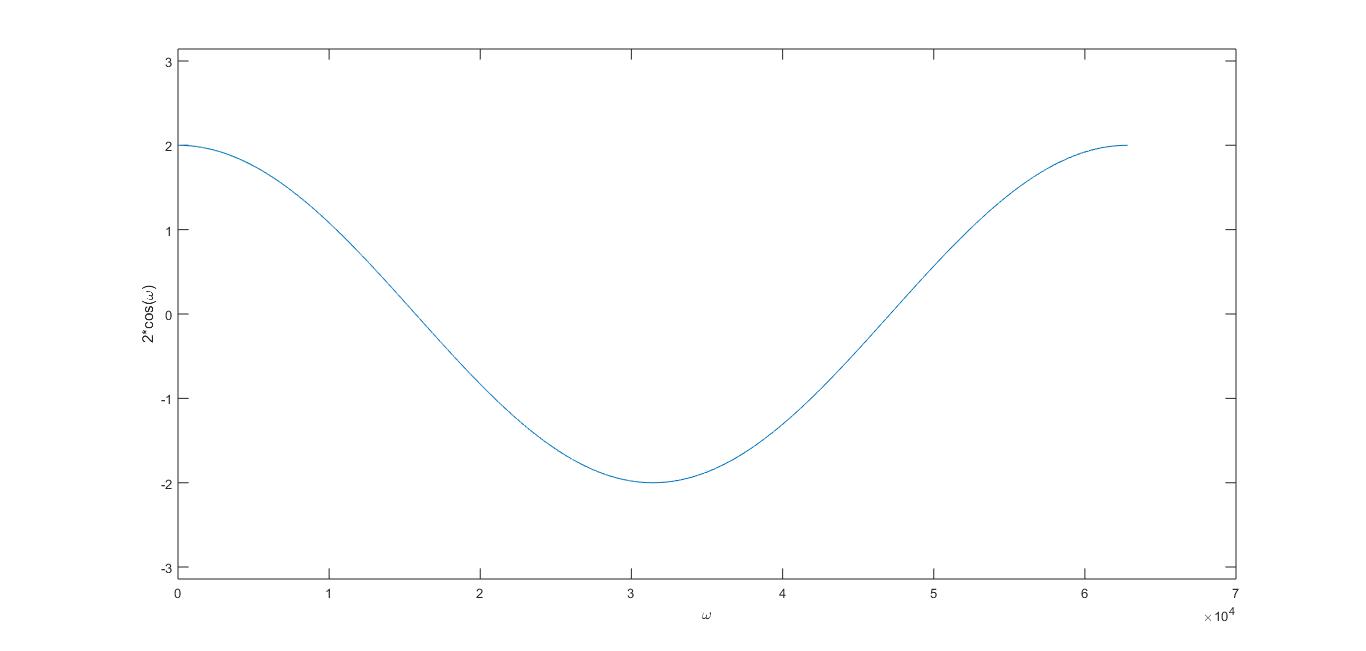
*xlabel('n');*

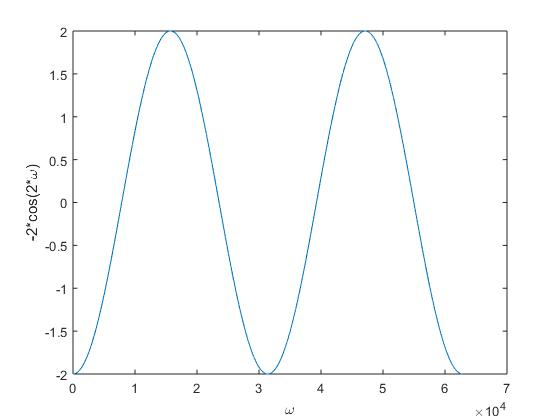
*ylabel('h(n)');*

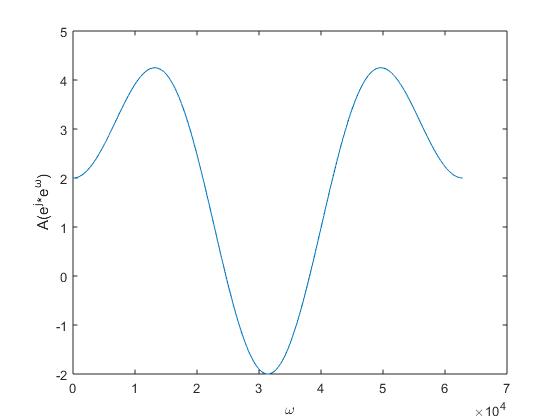


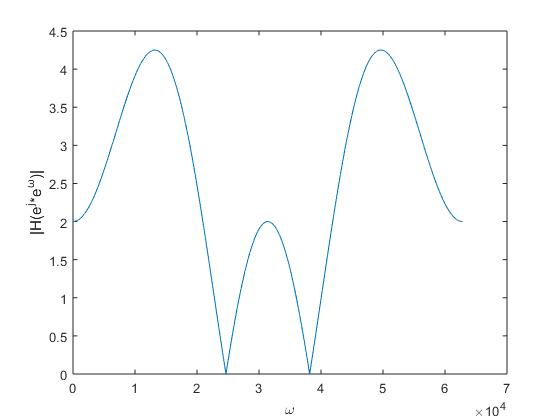
Hình 5.4.1.2

Đồ thị của và cho trên hình 5.4.1.3 và ta thấy rằng nếu h(n) đối xứng và N lẻ thì là đối xứng trong khoảng









*x= [0:0.0001:2\*pi];*

*y=2\*cos(x);*

*xlim([0 2\*pi]);*

*ylim([-2 2]);*

*plot(y);*

*figure*

*y=-2\*cos(2\*x);*

*plot(y);*

*figure*

*y=*

*plot(y);*

*figure*

*plot(abs(y))*

**7.1.3.2 Trường hợp đáp ứng xung đối xứng N chẵn (Bộ lọc FIR loại 2)**

Ta biết rằng:

N chẵn ta có thể chia tổng này thành 2 phần:

Đổi biến số ở thành phần thứ 2 ta có:

Đổi về cùng kí hiệu n và áp dụng tính đối xứng của h(n) là:

ta có:

ở đây:

1

So sánh với biểu thức:

Ta có:

Chú ý rằng với ⍵ = π thì:

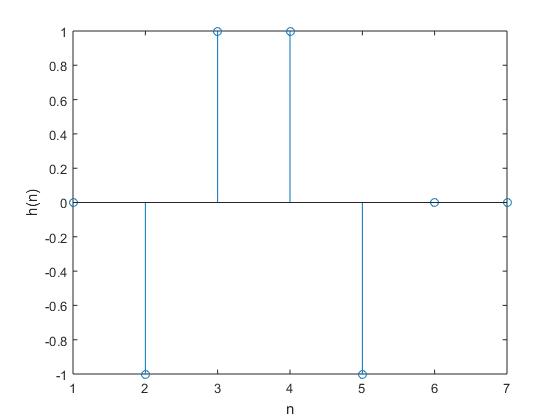
(2n – 1) là lẻ với mọi n, vậy:

với mọi n

Như vậy ta có thể nói rằng tại ⍵ = π thì với bất kì b(n) nào (hoặc là với bất kì h(n) nào), và từ đây ta rút ra kết luận là: các bộ lọc loại này không thể sử dụng để tổng hợp các bộ lọc có đáp ứng tần số khác không tại ⍵ = π (ví dụ như bộ lọc thông cao)

**Ví dụ 5.4.2.1 :**

*Cho đáp ứng xung của bộ lọc FIR pha tuyến tính như hình 5.4.2.1.*

**

Hình 5.4.2.1

*x=[0 -1 1 1 -1 0 0];*

*xlim([-1 5]);*

*ylim([-1 1]);*

*stem(x);*

*xlabel('n');*

*ylabel('h(n)');a*

*Hãy tìm b(n), và |*

**Giải**

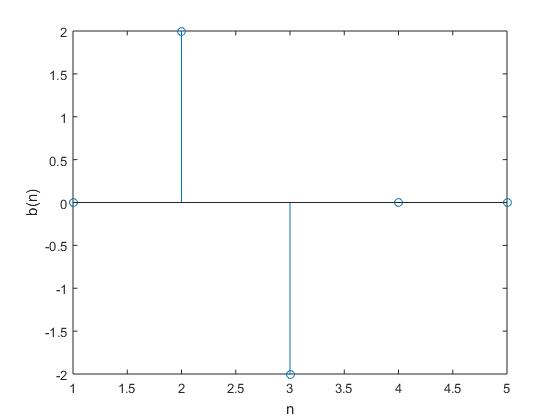
N=4 ,

Vậy:

b(1) = 2h(2-1) = 2h(1) = 2

b(2) = 2h(2-2) = 2h(0) = -2

Đồ thị của b(n) cho trên hình 5.4.2.2.



Hình 5.4.2.2

*x=[0 2 -2 0 0];*

*xlim([0 4]);*

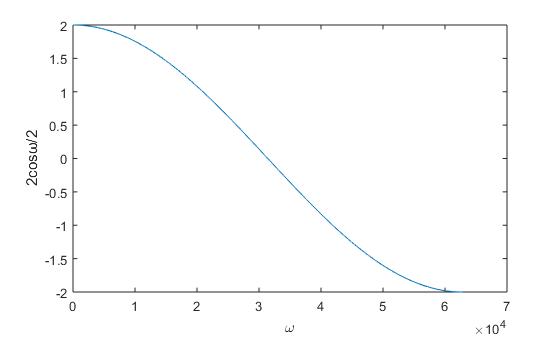
*ylim([-2 2]);*

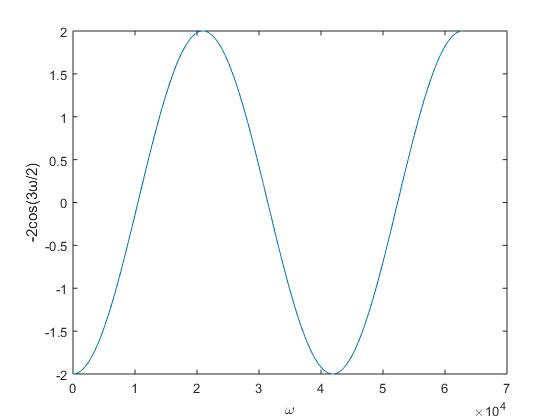
*stem(x);*

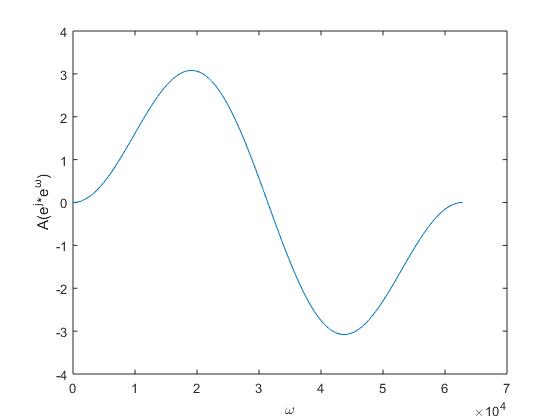
*xlabel('n');*

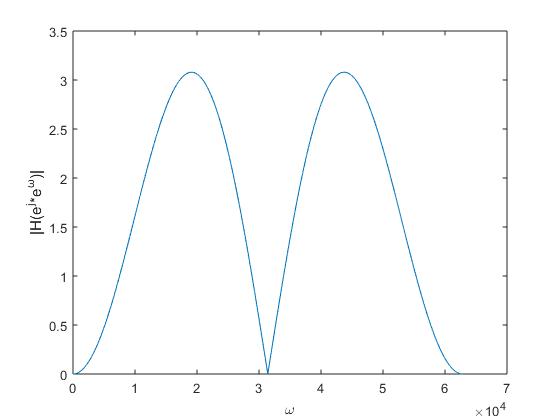
*ylabel('b(n)');*

Đồ thị của và cho trên hình 5.4.2.3, và ta thấy rằng nếu N chăn và h(n) đối xứng thì là phản đối xứng trong khoảng 0 ≤ ≤ 2π









Hình 5.4.2.3

*x=[0:0.0001:2\*pi];*

*y1=2\*cos(x/2);*

*y2=-2\*cos(3\*x/2);*

*y= 2\*cos(x/2) - 2\*cos(3\*x/2);*

*plot(y1)*

*figure*

*plot(y2)*

*figure*

*plot(y)*

*figure*

*plot(abs(y))*

**7.1.3.3 Trường hợp đáp ứng xung phản đối xứng, N lẻ (Bộ lọc FIR loại 3)**

Chúng ta đã có

N lẻ, ta phân tổng này thành 3 phần như sau:

Trong trường hợp này , vậy:

Đổi biến số ở thành phần thứ hai ta có:

Đổi về cùng kí hiệu n và áp dụng tính phản đối xứng của h(n) là:

ta có:

ở đây:

So sánh với biểu thức:

Ta có:

Nhận xét:

* Với ω = 0 và ω = π thì

sin ωn = sin 0n = 0 ∀n

sin ωn = sin πn = 0 ∀n

Như vậy ta có thể nói rằng tại ω = 0 và ω = π vơi bất kì c(n) nào (hoặc bất kì h(n) nào), và từ đây ta thấy rằng các bộ lọc loại này không thể dùng để tổng hợp các loại bộ lọc có đáp ứng tần số khác không tại ω = 0 và ω = π (ví dụ như bộ lọc thông thấp và thông cao).

**7.2 Phương pháp cửa sổ**

**7.2.1 Cửa sổ chữ nhật**

Trong miền cửa sổ chữ nhật được định nghĩa như sau:

Hình 7.2.1.1 dưới đây cho ta đồ thị của cửa sổ chữ nhật.

Hình 7.2.1.1 Đồ thị cửa sổ hình chữ nhật

Từ trên ta có thể thấy rằng cửa sổ chữ nhật cũng chính là dãy chữ nhật .

Cửa sổ này được sử dụng để hạn chế chiều dài của lý tưởng có pha tuyến tính. Cửa sổ là đối xứng, tâm đối xứng nằm tại ( lẻ) vậy trong miền tần số sẽ có pha tuyến tính là (FIR loại 1) thì trong miền , và sẽ có cùng tâm đối xứng tại ( lẻ). Như vậy phiên bản chiều dài hữu hạn của chiều dài vô hạn sẽ cho bởi biểu thức sau đây:

Tương đương:

**7.2.2 Cửa sổ tam giác (cửa sổ Bartlett)**

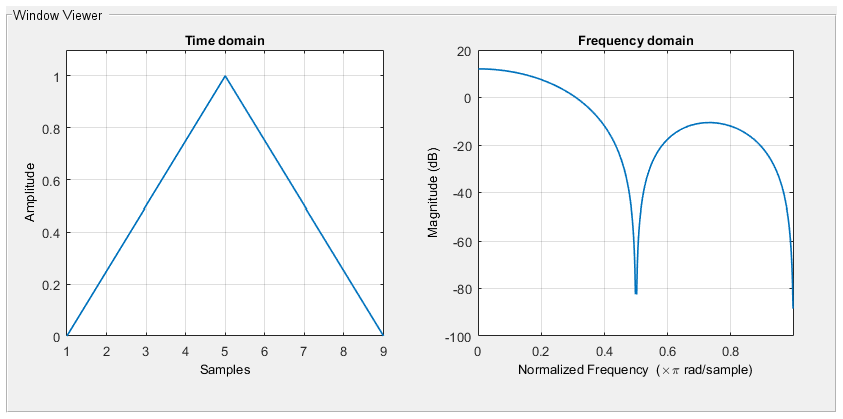
Với mục đích giảm biên độ của các đỉnh thứ cấp của cửa sổ chữ nhật , chung sta chọn một cửa sổ khác có dạng đồ thị tam giác cân, gọi là cửa sổ tam giác (cửa sổ Bartlett).

Trong miền cửa sổ tam giác được định nghĩa như sau:

*Ví dụ 7.2.2.1:* Biểu diễn đồ thị cửa sổ tam giác với .

Giải

Cửa sổ tam giác :



Hình 7.2.2.1 Đồ thị cửa sổ tam giác trên miền thời gian và đồ thị tần số

*Ví dụ 7.2.2.2:* Hãy tổng hợp bộ lọc thông thấp FIR pha tuyến tính dùng cửa sổ tam giác với , .

L = 9;

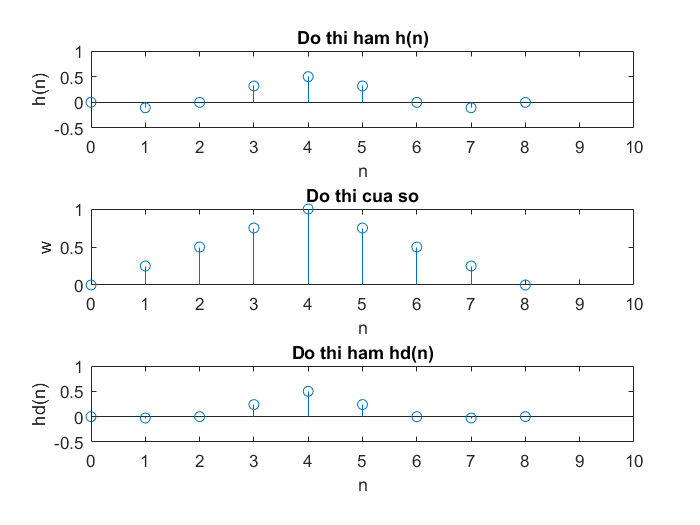
bw = bartlett(L);

wvtool(bw)

Giải

Bộ lọc thông thấp hữu hạn được tổng hợp theo biểu thức:

Thu được kết quả:



Hình 7.2.2.2 Đồ thị tín hiệu , và

wc = 0.5\*pi;

N = 9;

phi=(N - 1)/2;

n = 0:N-1;

w\_bart = bartlett(N);

h = (wc/pi)\*sin(wc\*(n-phi))./(wc\*(n-phi));

h(1,5) = 0.5;

hd = h.\*(w\_bart.');

subplot(3,1,1);

stem(n,h); title('Do thi ham h(n)');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('h(n)');

subplot(3,1,2);

stem(n,w\_bart); title('Do thi cua so');

axis([0 10 0 1]); xlabel('n'); ylabel('w');

subplot(3,1,3);

stem(n,hd); title('Do thi ham hd(n)');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('hd(n)');

**7.2.3 Cửa sổ Hanning và Hamming**

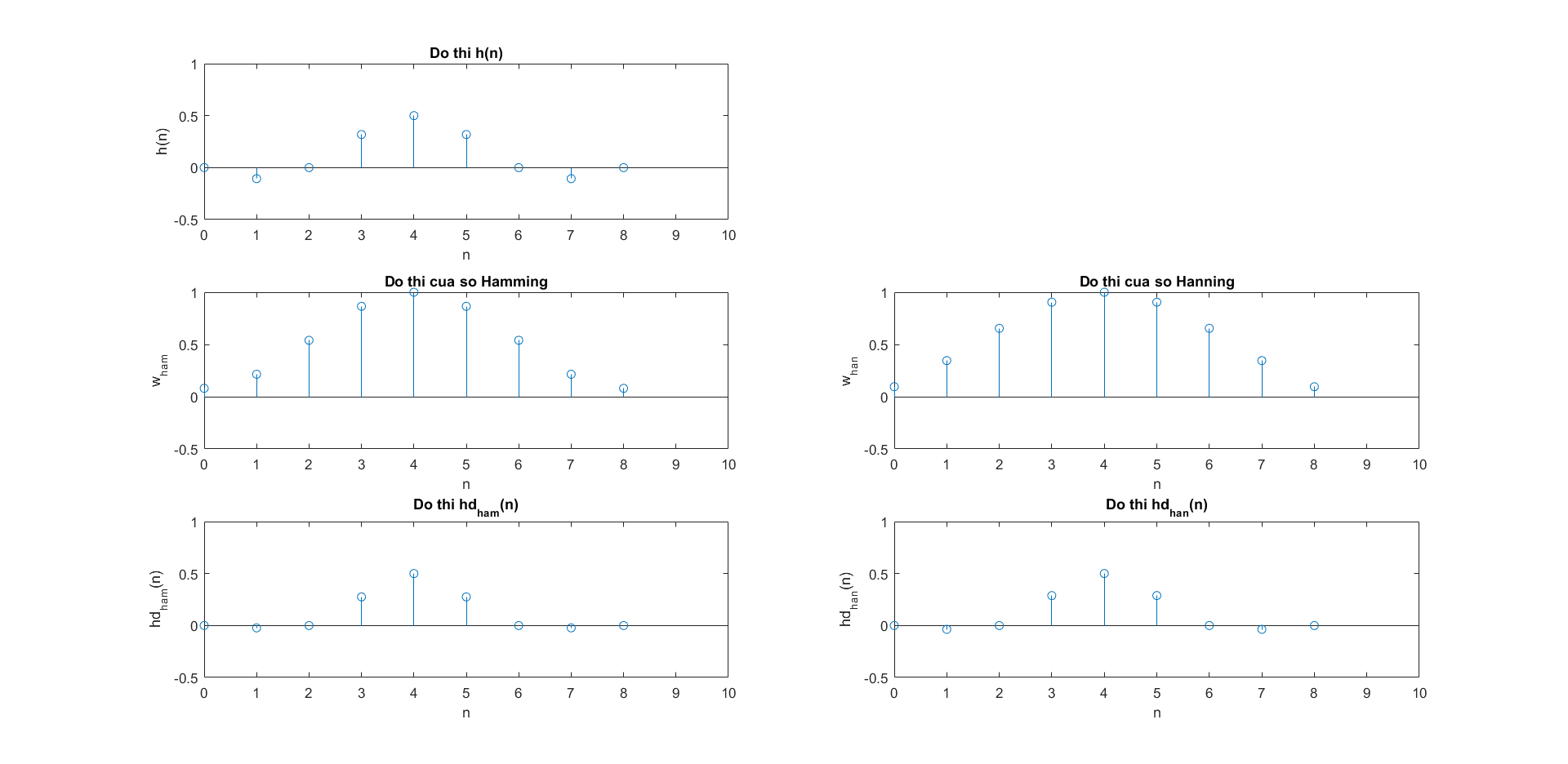
Trong miền cửa sổ Hanning và Hamming được định nghĩa như sau:

Nếu , ta có cửa sổ Hanning như sau:

Nếu , ta có cửa sổ Hamming như sau:

*Ví dụ 7.2.3.1:* Hãy tổng hợp bộ lọc thông thấp FIR pha tuyến tính dùng cửa sổ Hanning và Hamming với , .

Giải



Hình 7.2.3.1 Đồ thị tín hiệu, , và ,

wc = 0.5\*pi;

N = 9;

phi=(N - 1)/2;

n = 0:N-1;

w\_ham = hamming(N);

w\_han = hanning(N);

h = (wc/pi)\*sin(wc\*(n-phi))./(wc\*(n-phi));

h(1,5) = 0.5;

hd\_ham = h.\*(w\_ham.');

hd\_han = h.\*(w\_han.');

subplot(3,2,1);

stem(n,h); title('Do thi h(n)');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('h(n)');

subplot(3,2,3);

stem(n,w\_ham); title('Do thi cua so Hamming');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('w\_h\_a\_m');

subplot(3,2,4);

stem(n,w\_han); title('Do thi cua so Hanning');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('w\_h\_a\_n');

subplot(3,2,5);

stem(n,hd\_ham); title('Do thi hd\_h\_a\_m(n)');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('hd\_h\_a\_m(n)');

subplot(3,2,6);

stem(n,hd\_han); title('Do thi hd\_h\_a\_n(n)');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('hd\_h\_a\_n(n)');

**7.2.4 Cửa sổ Blackman**

Trong miền cửa sổ Blackman được định nghĩa như sau:

Điều kiện:

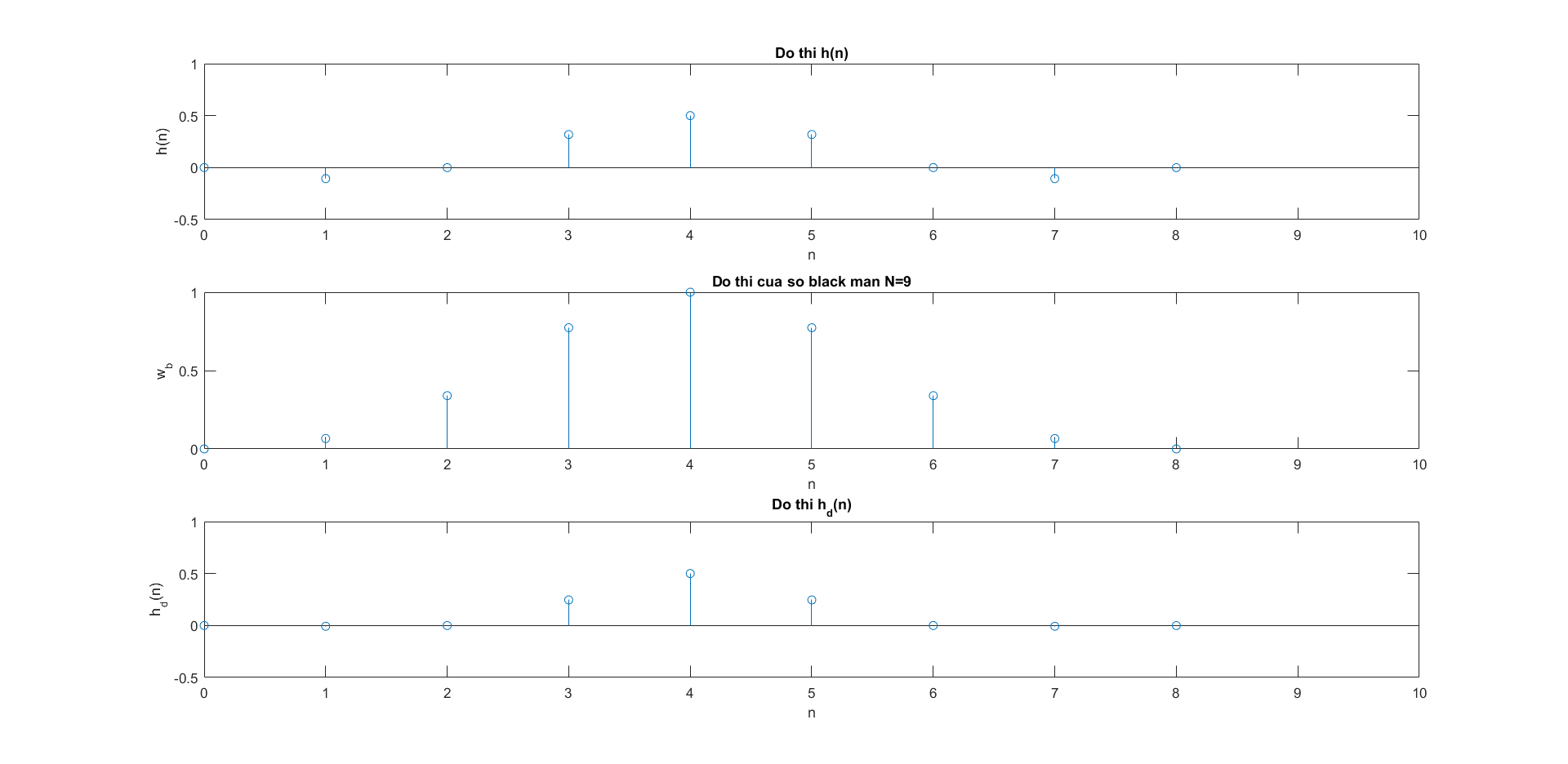
Cửa sổ có dạng trong biểu thức trên được gọi là cửa sổ Blackman tổng quát hóa. Ta thấy rằng các cửa sổ Hanning và Hamming chính là cửa sổ Blackman với hai hệ số và khác không:

với

Trong thực tế Blackman thường dung ba hệ số khác không là , và . Việc xác định các hệ số này với mục đich làm giảm bề rộng của đỉnh trung tâm ta tìm được:

Xuất phát từ các giá trị này , Blackman đã định nghĩa một cửa sổ, sau này gọi là cửa sổ Blackman như sau:

*Ví dụ 7.2.4.1:* Hãy tổng hợp bộ lọc thông thấp FIR pha tuyến tính dùng cửa sổ Blackman với , .



Hình 7.2.4.1 Đồ thị tín hiệu , và

vwc = 0.5\*pi;

N = 9;

phi=(N - 1)/2;

n = 0:N-1;

w\_blackman = blackman(N);

h = (wc/pi)\*sin(wc\*(n-phi))./(wc\*(n-phi));

h(1,5) = 0.5;

hd = h.\*(w\_blackman.');

subplot(3,1,1);

stem(n,h); title('Do thi h(n)');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('h(n)');

subplot(3,1,2);

stem(n,w\_blackman); title('Do thi cua so');

axis([0 10 0 1]); xlabel('n'); ylabel('w\_b');

subplot(3,1,3);

stem(n,hd); title('Do thi h\_d(n)');

axis([0 10 -0.5 1]); xlabel('n'); ylabel('h\_d(n)');

**7.2.5 Cửa sổ Kaiser**

Kaiser đã đưa ra một học các hàm cửa sổ. Trong miền tần số nó cho phép chúng ta định rõ mâu thuẫn giữa bề rộng của đỉnh trung tâm và biên độ của các đỉnh thứ cấp, mà chủ yếu là của đỉnh thứ cấp thứ nhất để xác định bằng cách thay đổi giá trị của tham số .

Một đặc trưng rất quan trọng của họ cửa sổ này là tạo ra khả năng giảm mạnh biên độ của các đỉnh thứu cấp mà vẫn đảm bảo bề rộng tối thiểu của đỉnh trung tâm.

Trong miền cửa sổ Kaiser được định nghĩa dạng tổng quát như sau:

Nếu càng lớn thì bề rộng của đỉnh trung tâm sẽ càng lớn.

Ở đây là hàm Bessel biến dạng loại một bậc không.

Bởi vì hàm Bessel rất phức tạp , vì vậy việc tính biểu thức giải tích của biến đổi Fourier là rất phức tạp và người ta không thể tìm một cách chính xác dưới dạng giải tích, nhưng người ta có thể tìm một cách gần đúng.

**7.3. Phương pháp lấy mẫu tần số**

- Xét đáp ứng xung của bộ lọc số thực tế FIR có chiều dài hữu hạn N:

* Biến đổi Fourier rời rạc với N điểm:
* Biến đổi Fourier rời rạc ngược:
* Xác định theo hàm của :
* chính là được đánh giá dựa trên vòng tròn đơn vị tại những điểm rời rạc với:

* Đối với đáp ứng tần số ta có:
* (3.3.3.1)

Với:

Biểu thức (3.3.3.1) chỉ đúng với dãy có chiều dài hữu hạn là N. Với bộ lọc số lý tưởng, đáp ứng xung có chiều dài vô hạn. Khi đó hàm truyền và đáp ứng tần số của bộ lọc số lý tưởng được xác định:

Xét gần đúng bằng hàm của bộ lọc thực tế. nhận được qua việc nội suy giữa các mẫu lấy trên tại các tần số

Sai số của phép gần đúng trên bằng không tại tần số và là hữu hạn đối với các tần số khác.

Tức là:

⬄

⬄

Qua phép gần đúng ta thu được bộ lọc số lí tưởng:

Bộ lọc số sẽ không lý tưởng nếu:

Với hàm nội suy:

Như vậy, ta có thể lấy từ N mẫu đáp ứng tần số của bộ lọc số lý tưởng để thu được đáp ứng tần số của bộ lọc số thực tế .

Thu được hai loại lấy mẫu tần số:

Loại 1:

Loại 2:

**7.3.1. Loại 1**

Xác định theo hàm của , trong tường hợp này , có thể viết:

Với bộ lọc số FIR pha tuyến tính thì:

Xét các bộ lọc cụ thể:

**7.3.1.1. Bộ lọc FIR N lẻ, đối xứng**

Ví dụ với N=9 ta thu được :

**7.3.1.2. Bộ lọc FIR N chẵn, đối xứng**

Ví dụ với N=8 ta thu được :

**7.3.2. Loại 2**

**7.3.2.1. Bộ lọc FIR N lẻ, đối xứng**

Ví dụ với N=9 ta thu được :

**7.3.2.2. Bộ lọc FIR N chẵn, đối xứng**

Ví dụ với N=8 ta thu được: