**Department of Computer Engineering**

**Faculty of Engineering, Kasetsart University**

### Program #03 FIR System

## **วัถตุประสงค์:**

ศึกษาและทำความเข้าใจในระบบ FIR

## **แนะนำเบื้องต้น:**

*จากบทเรียนที่ผ่านใน Time Domain เราสามารถจะแทนระบบหรือฟิลเตอร์ได้ใน* 2 *ร*ู*ปแบบคือ* ***difference equation*** *หรือ* **impulse response** *ซึ่ง* 2 *รูปแบบมีความแตกต่างกัน (แต่ในระบบ* FIR *นั้น) ในแบบฝึกหัดนี้นิสิต*

1. ***Difference equation*** *รูปสมการเต็มคือ*

…………………….(1)

*ซึ่งหากจัดเทอมใหม่ให้เอาต์พุตปัจจุบันคือ* *y[n]* *อยู่ด้านขวามือจะได้สมาการ*

……………(2)

*จากสมการจะเห็นได้ว่าการคำนวนหาผลลัพธ์ของระบบหรือฟิลเตอร์นั้นทำเพียงแค่การหาผลของของค่าอินพุตและเอาต์พุตที่มีน้ำหนักไม่เท่ากันนั้นเอง ซึ่งในสมการที่* 1 *สามารถระบุบได้ว่า* ***ai*** *คือค่าน้ำหนักประจำเอาต์พุตของ ในขณะที่* ***a0*** *น้ำหนักประจำเอาต์พุตของปัจจุบัน* และ ***bj*** *น้ำหนักประจำอินพุต ส่วนในสมการที่ 2 คือการจะจัดรูปใหม่เพื่อให้เอาต์พุตของปัจจุบันอยู่ด้านซ้ายมือ ดังนั้นจะเห็นว่าในกรณีนี้****a0*** =1

หากว่าค่าน้ำหนักของ*เอาต์พุตอื่นคือ* ***ai*** = 0 (ยกเว้น ***a0*** ซึ่ง*น้ำหนักประจำเอาต์พุตของปัจจุบัน) สมการจะเป็นระบบของ* FIR *และ* ***bj*** *ก็จะสามารถใช้แทน impulse response ของระบบ FIR ได้เลย (ในกรณีระบบ IIR จะต้องใช้วิธีคำนวณหา impulse response อีกทีหนึ่ง )*

*\*\*\*\*หากนิสิตได้เรียนต่อไปในเรื่องของ Z Transform ก็จะพบว่าค่า* ***ai*** *คือค่าสัมประสิทธิ์ของส่วน(denominator coefficients) และค่า* ***bj*** *คือค่าสัมประสิทธิ์ของเศษใน System Function นั้นเอง*

*ใน Matlab มีฟังก์ชันให้เราเรียกใช้ในการคำนวณดังนี้*

**Y =filter(b,a,x);** *คือฟังก์ชันใช้คำนวณผลลัพธ์ของสัญญาณเมื่ออกจากระบบหรือ Filters สัญญาณ ซึ่งสัญญาณเข้า (discrete signal) คือ* ***x*** *ส่วนค่า* ***a*** *และ* ***b*** *คือ vector ของสัมประสิทธิ์ในสมการ difference equation (ในกรณีของระบบ FIR นั้นเราใช้* ***a*** *คือ 1) สัญญาณผลลัพธ์จะมีจำนวนเท่าจำนวนสัญญาณขาเข้า*

**[H T]=impz (b, a, N);** *คือฟังก์ชันหาค่า* impulse response *ของระบบ ส่งคืนตัวแปร H ส่วน T จะเป็นค่าลำดับ ซึ่งค่า* ***a*** *และ* ***b*** *คือ vector ของสัมประสิทธิ์ในสมการ difference equation และควรระบุค่า N ในกรณีของ (ในกรณีของ FIR เราใช้* ***a*** *คือ 1 ล้วจะได้ผลลัพธ์ที่ใน H จะเท่ากับ b ตามที่เราเรียนมาแล้ว)*

**[H, f]=freqz(b,a,N,Fs);** *วาดกราฟ H* *ของ* *frequency response* *ของระบบ Filter โดย* ***a*** *และ* ***b*** *คือ vector ของสัมประสิทธิ์ในสมการ difference equation ถ้าระบุค่า Fs (*sampling frequency*) และ N ค่า*จำนวนเชิงซ้อน *H จะถูกคำนวณตาม* ***N*** *จุดที่ที่ความถี่* ***f*** *ซึ่งได้จาก Fs (sampling frequency). แต่ถ้าไม่ระบุ* ***Fs*** *ความถี่ที่ได้จะความถี่เชิงมุมใช่จำนวนรอบต่อวินาที (Hz)*.

C = conv(A, B) *การผลลัพธ์ของการ convolve ระหว่าง vectors A และ B. เราสามารถใช้ฟังก์ชั่นนี้หาผลลัพธ์สัญญาณของระบบหาเรา แทนด้วย A ด้วย h หรือ impulse response* และแทน B ด้วยสัญญาณขาเข้า vector ผลลัพธ์จะมีความยาวเท่ากับ

MAX([LENGTH(A) + LENGTH(B)-1,LENGTH(A),LENGTH(B)])

*ตัวอย่างเช่นใน FIR filter "Average filter" ซึ่งสามารถแสดงสมการ difference equation ได้ดังนี้*



*จากสมการการ เราทราบว่า a0 = 1และ bk = [1/3 1/3 1/3]*

*เราสามารถหาค่า h หรือ impulse response ของระบบจากฟังชั่น Matlab ดังนี้*

*h=impz( [1/3,1/3,1/3] ,1)*

*ซึ่งจะได้* h(n)= []

*ดังนั้นหากเรามีสัญญา x เราก็จะสามารถหาผลลัพธ์สัญญาณที่ออกจากระบบได้จากทั้งฟังก์ชั่น filter และ conv (แต่ต้องระวังผลลัพธ์จาก conv จะมีจำนวนมากเกิน)*

**LAB & QUESTIONS**

1. ***Average Filter:***

*สมมุติให้สัญญาณอินพุตเริ่มต้นเป็นศูนย์หมด x[n]=0 for n≤0*

1. *ให้แสดงการหา impulse response ที่เราเรียนในห้อง (impulse response คือ เอาท์พุตของระบบที่เกิดจากอินพุตที่เป็น impluse*)

2. *สร้างสัญญาณอินพุต* ***x(n)*** *ซึ่งมีสัญญาฯรบกวนผสมแบบ exponential noise แล้วจงวาดสัญญาณที่ได้ออกมา* *เปรียบเทียบ ให้ใช้คำสั่ง subplot วางสัญญาณ s*

R = 51;

d = randi(50,1,R)/100

m = 0:R-1;

s = 2\*m.\*(0.9.^m); % Generate uncorrupted signal

x = s + d; % Generate noise corrupted signal

b=ones(5,1)/5;

a=1;

y= filter([1/3,1/3,1/3],1,x); เติมในวงเล็บให้ครบ

subplot(3,1,1);plot(m,s);

subplot(3,1,2);plot(m,x);

subplot(3,1,3);plot(m,y);

4. *จากนั้นให้คำนวณหาเอาท์พุตจากฟังก์ชั่น* ***conv*** *แล้วเปรียบผลระหว่าง* ***filter*** *มีอะไรที่แตกต่างกันและจะแก้ไขอย่างไร*

h=b

yc=conv(b,x);

size(yc);

ขนาดของconv จะมากกว่า ขนาดของ filter

แก้ไขโดยการเพิ่มขนาด mหรือตัด yc ส่วนเกินทิ้ง

*5. เมื่อไรเราจึงจะใช้ filter และ conv ในการประมวลผลสัญญาณ*

*Filter ในการคำนวณ vector A,B ,conv หา vector*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

***Real-Life Application: Stock value***

1. *ข้อมูลในไฟล์ stock.mat* *นี้คือราคาของหุ้นบริษัท* *(*Microsoft stock price*)* *ย้อนหลัง 4 ปี*

2. *ส่วนใหญ่ในการดูหุ้นเราจะมักจะเห็นการเคลื่อนไหวของราคาหุ้นขึ้นๆลงๆ*

3. *ให้นิสิตใช้คำสั่ง* Load *เอาข้อมูลในไฟล์ไว้ในตัวแปร* (*นิสิตอาจจะใช้คำสั่ง whos เพื่อตัวตัวแปรที่เกิดขึ้น*)

load('stock.mat');

whos

plot(stock(1,: ),stock(2,: ),'b');

hold;

เติมคำสั่งให้ครบจากข้อ 4 และ 5

h = ones(30,1)/30;

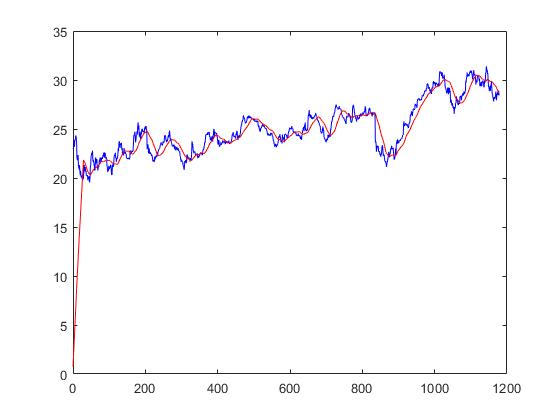
y = filter(h,1,stock(2,: ));

plot(stock(1,: ),y,'r');

4. *จากรูปที่ plot ได้จะเห็นราคาหุ้นเคลื่อนไหวอย่างรวดเร็วเกินไป ซึ่งโดยปกติการดูราคาหุ้นในภาพรวมมันนิยมดูแนวโน้มของราคาหุ้นมากกว่าในรายละเอียดแต่ละวัน (*long term trend*)* *ดังนั้นเราจะใช้* moving average filter *กรองข้อมูลราคาหุ้นก่อนการ* plot *นี้คือ* impulse response *ของ* moving average 30 *วัน*

5. *ให้นิสิตสร้าง* impulse response(hint: using ***ones*** command with weight). *จากนั้นให้กรองข้อมูลราคาหุ้น (สามารถใช้ได้* **conv** *และ* **filter**) *ให้* plot *ราคาหุ้นที่ถูกกรองบนกราฟเดียวกันด้วยสีแดง นิสิตเห็นราคาหุ้นใหม่มีความเรียบมากขึ้นหรือไหม มีปัญหาอะไรเกิดขึ้นกับกราฟที่ plot ได้อย่างไรบ้าง (ให้นิสิตสังเกตช่วงแรกๆ และให้สั่งราคาหุ้นใหม่จะช้าไป 30 วันเพราะ filter ที่สร้างขึ้นคือ causal filter)*

วางรูปที่ได้จากการทำงาน



1. ***LTI System***

*กำหนดให้* digital filter *มีคุณลักษณะ difference equation ดังนี้*



*สมมุติ* all zero initial conditions *นั้นคือ*  y[n]=0 *สำหรับ* n≤0

1. Filter *ที่กำหนดคือ* FIR *หรือ* IIR *และเพราะอะไร*?

FIR เพราะ มี y มาร่วมคำนวณด้วย

2. *จงสร้างสัญญาณ* ***x[n]*** *ได้จากผลรวมของสัญญาณ* sinusoids 2 *ความถี่ คือ* 300 Hz *ที่ขนาด 5และ* 1200 Hz Hz *ที่ขนาด 7* *โดย* sampled *ที่ความถี่* 6 kHz. *ให้วาดกราฟแสดง* 10 *ลูกคลื่นของสัญญาณ* 300 Hz

ใส่คำสั่งให้ครบ

samp\_rate = 6000

t1 = numpy.arange(0, 10/300, 1/samp\_rate)

x1 = 5\*numpy.cos(2\*math.pi\*300\*t1) + 7\*numpy.cos(2\*math.pi\*1200\*t1)

plt.plot(x1)

plt.title("x")

plt.show()

3. *จงหาค่าสัญญาณเอาท์พุต ใช้ฟังก์ชั่น* filter *ที่กำหนดให้ข้างต้น แล้วให้ plot สัญญาณต้นฉบับเทียบกับสัญญาณเอาท์พุต ให้สังเกตแล้วตอบว่า นิสิติคิดว่าระบบนี้ทำหน้าที่อะไร*

*b = [0.692, -1.762, 3.545, -3.925, 3.545, -1.762, 0.692]*

*a = [1, -2.238, 3.921, -3.844, 3.075, -1.367, 0.478]*

*y1 = scipy.signal.lflter(b, a, x1)*

*plt.plot(y1)*

*plt.title("y1")*

*plt.show()*

*คำตอบ …*

*กรอกเฉพาะความถี่ 300 Hz*

4. *ให้นิสิต plot ค่า frequency response ของระบบด้วยคำสั่ง freqz นิสิตสังเกตกราฟ และยืนยันคำตอบในข้อ 3 ให้ชัดเจน*

H, W = scipy.signal.freqz(b, a);

plt.plot(numpy.abs(H), (W\*6000)/(2\*math.pi))

plt.xlabel('Frequency in Hz');

plt.show()

*ยืนยันคำตอบ …ข้อ 3*

5. *ให้นิสิตคำนวณหา* impulse response *ของระบบด้วยฟังก์ชั่น* **impz** *ซึ่งค่าที่ได้คือ h ให้นิสิตใช้คำนวณหาสัญญาณเอาท์พุตด้วยฟังก์ชั่น conv แล้ว plot กราฟดู*

N = 100

x = scipy.zeros(N)

x[0] = 1

y2 = scipy.signal.convolve(h, x1)

plt.plot(y2[:200])

plt.title("y2")

plt.show()

*คำตอบ …*

|  |  |
| --- | --- |
| กราฟ x | กราฟ abs(H) |
| กราฟ y1 | กราฟ y2 |

***คำถาม*** *สัญญาณเอาท์พุตมีลักษณะอย่างไรเมื่อเทียบกับผลที่ได้จาก filter*

***คำตอบ ไม่ต่างกัน***

6. *การทำงานของ* filter *นี้* stable *หรือไม่*? *(นิสิตจะสังเกตได้ว่า filter จะให้ผลลัพธ์ในช่วงแรกๆ แปลกๆ แต่หลังจากนั้น filter จะทำงานอย่างสม่ำเสมอ เราเรียกสภาพนี้ว่า ระบบนี้มีความเสถียร* stable*)*

*คำตอบ …stable*

7. *นิสิตสังเกตเห็นว่า h หรือ impulse response มีจำนวนเท่าไร? จะเกิดอะไรขึ้นถ้าเราลดจำนวน h ลง เหลือเพียง 50 ค่า? (การลดค่าคือการประมาณ filter เพราะยิ่งจำนวน h น้อยลง filter จะประมวลผลได้เร็วขึ้น) แล้วลองลดจำนวนให้เหลือเพียง 32 ค่า นิสิตสังเกตผลลัพธ์ (การประมาณค่า h นิสิตจะได้เรียนรายละเอียดในบทของ IIR) ต่างกันหรือไม่อย่างไร และเพราะอะไร*

h50=h(0:50);

h32=h(0:32);

y3 = scipy.signal.lflter(h50, 1, x1) % you can use conv, but remember your output would have more %member than member of input

y4 = scipy.signal.lflter(h32, 1, x1)

plt.subplot(131)

plt.plot(y1)

plt.subplot(132)

plt.plot(y3)

plt.subplot(133)

plt.plot(y4)

plt.show()

*คำตอบ….*

ไม่ต่างกันเพราะhหลังตัวที่ 30 มีค่าประมาณ0

11.  *ให้นิสิตลองใช้ฟังก์ชั่น* **freqz** *เพื่อหา* frequency response *ของทั้ง h h50 h32 แล้วให้ plot กราฟเทียบกันดู filter ทั้ง 3 ทำหน้าที่ต่างหรือไม่*

[H50 W]= freqz(h50,1);

[H32 W]= freqz(h32,1);

figure;

subplot(3,1,1); plot((W\*6000)/(2\*pi),abs(H));xlabel('Frequency in Hz');

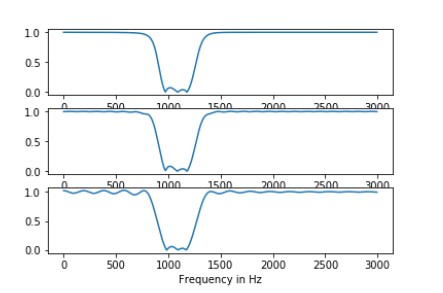
subplot(3,1,2); plot((W\*6000)/(2\*pi),abs(H50));xlabel('Frequency in Hz');

subplot(3,1,3); plot((W\*6000)/(2\*pi),abs(H32));xlabel('Frequency in Hz');

*คำตอบ….*

*ไม่ต่างกัน*

กราฟที่ได้



1. ***Digital Image Filter (LTI Convolution in digital image processing)***

**Convolution in Digital Image processing**

*Linear filtering บนภาพถ่ายดิจิตอล ก็สามารถทำได้โดยวิธี convolution ซึ่งการทำ Convolution คือการผลรวมของผลคูณระหว่างค่าความสว่างของแสงในตำแหน่ง pixel ที่ตรงกับค่าน้ำหนักหรือ weight ของ h ซึ่งจะเรียกว่า* convolution kernel (นิสิตอย่าลืมการกลับทิศของระบบหรือ Filter ซึ่งกรณี filter เป็น 2 มิติจะต้องกลับทิศ 2 ครั้ง เท่า 180 องศา)

ตัวอย่างเช่น subimage

A = [17 24 1 8 15

23 5 7 14 16

4 6 13 20 22

10 12 19 21 3

11 18 25 2 9]

*และ* convolution kernel *คือ*

h = [8 1 6

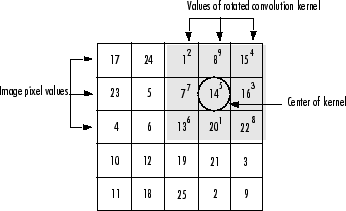
3 5 7

4 9 2]

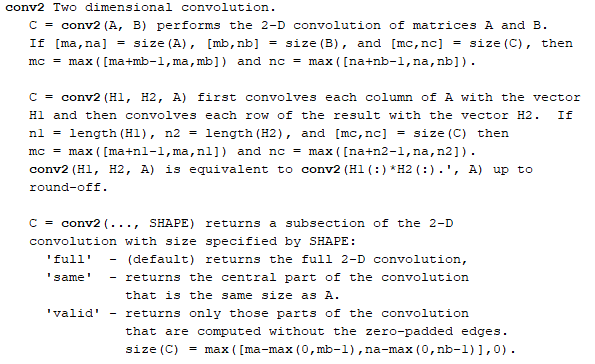
*ขั้นตอนการคำนวณหาค่า pixel ผลลัพธ์ที่ตำแหน่ง (2,4) ทำดังนี้:*

1. Rotate the convolution kernel 180 degrees about its center element.
2. Slide the center element of the convolution kernel so that it lies on top of the (2,4) element of A.
3. Multiply each weight in the rotated convolution kernel by the pixel of A underneath.
4. Sum the individual products from step 3.

linfilt14



**Performing Linear Filtering of Images Using conv2**



*การ Filtering รูปถ่ายดิจิตอล โดบใช้วิธี convolution นั้นใน Matlab สามาถใช้ ฟังก์ชั่น conv2 ได้ ดังตัวอย่าง* *เป็นการใช้ filter ขนาด 5-by-5 filter ที่เราเรียกว่า moving averaging filter*

I = imread('coins.png');

I = double(I); % since pixel value in ‘coins.jpg’ is unsigned integer

h = ones(5,5) / 25;

%h = rot90(rot90(h)); % do not need to rotate

I2 = conv2(double(I), double(h)); % 2D-convotion operation needs

%double value to compute

imshow(uint8(I)), title('Original Image'); % Covert back to unsigned integer

%before display

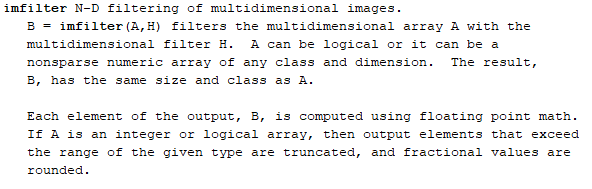
figure, imshow(uint8(I2));

title('Filtered Image by Convolution')

วางภาพผลลัพธ์

|  |  |
| --- | --- |
|  |  |
| Original Inage | Filtered Image |

**Performing Linear Filtering of Images Using imfilter**



*อีกวิธีหนึ่งในการทำ Filter ใน Matlab สามารถใช้ฟังก์ชั่น imfilter ได้โดยตรง (นิสิตจะต้องติดตั้ง image toolbox ของ Matlab ถึงจะใช้คำสั่งนี้ได้)*

I = imread('coins.png');

I = double(I); % since pixel value in ‘coins.jpg’ is unsigned integer

h = ones(5,5) / 25;

I3 = imfilter(I,h);

figure; imshow(uint8(I3)), title('Filtered Image using imfilter ')

วางภาพผลลัพธ์

|  |  |
| --- | --- |
|  |  |
| Original Inage | Filtered Image |

คำถาม

1. *มีอะไรแตกต่างระหว่างภาพผลลัพธ์ของทั้ง 2 วิธี ให้นิสิตสังเกต เช่นขนาดของภาพ และทำไมถึงเป็นเช่นนั้น*

ตอบ สีของภาพต่างกัน เพราะวิธีการจัดการอาร์เรย์แตกต่างกัน

1. *ทำไมในฟังก์ชั่น conv2 เราจำต้องกลับทิศ Filter 180 องศา (ทำการหมุ่นในฟังก์ชันให้เอง) เพราะเหตุอะไร*

ตอบ index อาร์เรย์เริ่มต้นที่มุมบนซ้าย แต่ภาพ อาร์เรย์ (0, 0) อยู่ล่างซ้าย

1. *กรณีนี้ที่เราใช้ Filter แบบ moving averaging filter ขนาด 5-by-5 filter ในฟังก์ชั่น conv2 และ imfilter ยังต้องกลับทิศอีกไหม เพราะอะไร (นิสิตต้องตรวจดูค่าภายในของ h ก่อนจึงจะตอบคำถามนี้ได้ดี)*

ตอบ ไม่ต้องกลับทิศ array สมมาตรอยู่แล้ว