**Department of Computer Engineering**

**Faculty of Engineering, Kasetsart University**

**Program #04 Frequency response and Filter type**

**วัถตุประสงค์:**

ในการทดลองนี้นิสิตจะทบทวนและเรียนรู้เพิ่มเติมเกี่ยวกับ Frequency response ซึ่งเป็นพื้นของการออกแบบ Filter ทั้ง FIR และ IIR

* **Frequency Response ด้วยฟังก์ชั่น freqz (ทบทวน)**



* Freqz เป็นฟังก์ชั่นที่ใช้เพื่อดูการตอบสนองของ Filter เมื่ออินพุตคือสัญญาณ sinusoid ความถี่ใดๆ ผลที่ได้จากฟังกชั่นคือ complex number ที่แสดงถึง transfer function ของระบบหรือ filter
* การ plot การตอบสนองของความถี่ จะทราบ difference equation คือ a และ b แต่ในระบบ FIR เราทราบมาแล้วว่า a = [1] และยิ่งไปกว่านั้นเราทราบทันทีว่า impulse response หรือ h ซึ่งคือ b นั้นเอง
* ในการ plot การตอบสนองของความถี่ แกน x จะเป็นความดิจิตอลเชิงมุมเสมอ และเรานิยม plot แค่ 0 ถึง π (ครึ่งวงกลมด้านบนของ Unit circle) หรือ 0 ถึง fs/2 (ความถี่ที่เป็นลบเราไม่ใส่ใจเพราะไม่มีอยู่จริง มันถูกใช้แค่การคำนวณเท่านั้น แต่อยากจะให้ plot ถึงลบ π ต้องเพิ่มพารามิเตอร์ 'whole'). ถ้าเรา plot ถึง 2π เราจะพบว่ามันซ้ำเป็นรายคาบ แต่เราก็ไม่สนใจช่วงเลยไปจาก π เพราะมันคือ Alias frequency ซึ่งจะไม่มีตัวตนจริงแต่เก็บเป็น principle frequency แทน
* ค่าของ a และ b ใน difference equation เป็นกำหนดการกรองสัญญาณเข้าว่าจะให้กันหรือขยายสัญญาณที่ความถี่ใด
* **ชนิดของ Filter แบ่งตามการกรองสัญญาณตามย่านความถี่**
* เราแบ่ง Filter ได้ 4 แบบที่แตกต่างกัน ซึ่งเราจะใช้บ่อยมากในการทำงานทั่วไปของ DSP
* Lowpass filter (LPF)
* Highpass filter (HPF)
* Bandpass filter (BPF)
* Bandstop filter (BSP)
* เราแยกประเภท Filter นี้ได้อย่างไร?

จากรูปถัดไปเราจะเห็นลักษณะของกราฟ ซึ่งสังเกตจากการตอบสนองของ Filter กับความถี่ของสัญญาณอินพุต ที่เราเรียกว่า Frequency Response คือ การขยายหรือการกดลงต่ำ(กรอง) ของสัญญาณในแต่ละย่านความถี่หรือความสูงต่ำของแต่ละย่านความถี่ ดังนั้นเราจะวาดกราฟ Frequency Response ด้วยฟังก์ชั่น freqz ของ Matlab

* แล้วเราทราบได้อย่างไร?

ถ้านิสิตสังเกตุที่แกนนอน ของ กราฟ Frequency Response ความถี่ จิติทอล เชิงมุม หรือ ที่เราเรียนกัน โดยแกนนอนเรานิยมวาดกราฟตั้งแต่ 0 ถึง π ซึ่งเราทราบว่าที่ตำแหน่ง π ความถี่สูงสุดที่มีในระบบจิติทอล ดังนั้นช่วงใกล้ 0 เราทราบทันทีว่าคือสัญญาณย่านความถี่ต่ำ แต่ถ้าช่วงใกล้ๆ π คือย่านความถี่สูง

* ลักษณะของ Filter ที่สังเกตุจากกราฟ Frequency response เป็นลักษณะทั่วไปประจำตัว Filter ดังนั้นเมื่อเราจะใช้งาน Filter เราต้องทราบ ความถี่ของการสุ่ม (Sampling rate) เพื่อจะทราบว่า Filter จะกรอง/ขยายที่ความถี่ใดบ้าง จากความสัมพันธ์ของ

 หรือ 



|  |  |
| --- | --- |
| LPF= Low pass filter  ขยายสัญญาณย่านความถี่ต่ำแต่กดสัญญาณย่านความถี่สูง | HPF= High pass filter  ขยายสัญญาณย่านความถี่สูงแต่กดสัญญาณความย่านถี่ต่ำ |
| BPF=Band pass filter  ขยายสัญญาณในช่วงย่านความถี่หนึ่ง แต่กดสัญญาณย่านความถี่อื่นๆ | BSF=Band stop filter  กดสัญญาณในช่วงย่านความถี่หนึ่ง แต่ขยายสัญญาณย่านความถี่อื่นๆ |

**การทดลอง**

* **การสร้าง Frequency Response จาก Differential equation**
* วิธีหนึ่งของการอธิบายการทำงานของระบบ LTI คือบอกความสัมพันธ์ระหว่างการคำนวณหาเอาต์พุต y[n] จาก อินพุต x ต่างๆ ตัวอย่างเช่น

*y[n] - 2 y[n - 1] = x[n] + 5x[n - 1]*

หรือ

*y[n] = x[n] + 0.5x[n - 2] - 0.7 y[n - 1] + 0.2 y[n - 1]*

* เราสามารถจัดเทอมที่เกี่ยวกับเอาต์พุต y และ อินพุต x ไว้ด้วยกันโดยให้เอาต์พุตอยู่ซ้ายมือ และอินพุตขวามือ



* แยก coefficients เป็นกลุ่ม B และ A โดย B คือ coefficients ของอินพุต x และ A coefficients ของเอาต์พุต y

จากตารางที่ให้ด้านล่าง จงแยกกลุ่ม coefficients A และ B ใส่ไว้ในตาราง ช่องขวามือ

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Differential Equation** | **B and A coefficients** | |
| 1 | *y*[*n*] - 2 *y*[*n* -1] = *x*[*n*] + 5*x*[*n* -1] | B = [1  A = [1 | 5]  -2] |
| 2 | *y*[*n*] = *x*[*n*] + 0.5*x*[*n* - 2] - 0.7 *y*[*n* -1] + 0.2 *y*[*n* -1] | B = [1 0 0.5]  A = [1 0.7 -0.2] | |
| 3 | *y*[*n*] = *x*[*n*] - *x*[*n* -1] | B = [1 -1]  A = [1] | |
| 4 | *y*[*n*] = *x*[*n* +1] - *x*[*n*] | B = [1 -1]  A = [0] | |
| 5 | *y*[*n*] - *y*[*n* - 2] = *x*[*n*] + 2*x*[*n* - 2] | B = [1 0 2]  A = [1 0 -1] | |
| 6 | *y*[*n*] - *y*[*n* -1] + 0.5 *y*[*n* - 2] = *x*[*n*] | B = [1]  A = [1 -1 0.5] | |
| 7 | *y*[*n* +1] - 0.5 *y*[*n* - 2] = *x*[*n* -1] - 2*x*[*n* - 2] | B = [0 0 1 -2]  A = [1 0 0 -0.5] | |

* ให้นิสิตใช้ฟังก์ชั่น ‘freqz’ หา frequency response จาก Differential equation พร้อมระบุ ชนิดของFilter ใส่ในตาราง

|  |  |  |
| --- | --- | --- |
| **No.** | **กราฟของ frequency response** | **ชนิดของ Filter** |
| 1 |  | Low Pass |
| 2 |  | Band Stop |
| 3 |  | High Pass |
| 4 |  | High Pass |
| 5 |  | Band Stop |
| 6 |  | Band Pass |
| 7 |  | Band Stop |

**คำถาม** กราฟที่ได้จาก Frequency Response หรือจากฟังก์ชั่น freqz ในแกนนอนคือความถี่จิติทอลเชิงมุม หรือ และถ้าเราทราบถึง *fs* เรามีวิธีอะไรที่จะให้แสดงค่าในแกนนอนให้แสดงความถี่สัญญาณแอนนาลอก *f* ได้อย่างไร จงใช้ filter ในข้อ 2 ของตาราง ที่ถูกนำใช้ filter กรองสัญญาณความถี่ที่ถูกสุ่มมาด้วย sampling rate 2 KHz แสดงคำสั่งของ Matlab ในคำตอบ พร้อมกราฟที่ได้

คำตอบ

เปลี่ยนจาก แกนนอนเป็น w เป็น (w\*f)/(pi)

* **การหา Frequency response จาก impulse response**
* ในการณีของ FIR ถ้าพิจารณาจาก difference equation เราทราบว่า a = [1] และ impulse response หรือ h ซึ่งคือ b นั้นเอง จงหา Frequency response ของแต่ filter ในตารางถัดไป ซึ่งอธิบายระบบมาเพียงแค่ และ impulse response หรือ h และนำกราฟที่ได้ใส่ในตาราง พร้อมระบุชนิดของ Filter

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **h** | **กราฟ** frequency response | **ชนิดของ filter** |
| **1** | **[0.0284 0.2370**  **0.4692 0.2370**  **0.0284]** |  | **Low Pass** |
| **2** | **[0.0338 -0.2401**  **0.4521 -0.2401**  **0.0338]** |  | **High Pass** |
| **3** | **[0.0000 -0.1191**  **0.0000 0.7617**  **0.0000 -0.1191**  **0.0000]** |  | **Band Pass** |
| **4** | **[0.0000 0.1597 0 0.6806 0.0**  **0.1597 0.0000]** |  | **Band Stop** |
| **5** | **[0.5 0.5]** |  | **Low Pass** |
| **6** | **[0.5 -0.5]** |  | **High Pass** |

* **การประมาณการ filter แบบ IIR ให้ไปเป็น FIR**

หากนิสิตยังจำได้จากการทดลองครั้งก่อนว่า เราสามารถประมาณการ filter แบบ IIR ให้ไปเป็น FIR ได้ด้วยฟังก์ชั่น‘impz’ ซึ่งคำนวณค่า ‘*h*’ impulse response มาให้ ในเกือบทุกกรณี h ที่ได้จากการแปลงจะมีจำนวนมากไม่จำกัด ดังนั้นเวลาใช้งานเราจึงจำกัดไว้เพียงที่ต้องการ (เราจะเรียนการประมาณการ filter แบบ IIR ให้ไปเป็น FIR ด้วยทฤษฎีในบทของ IIR Filter อีกครั้ง)

***[h,t] = impz(b,a)***

returns the impulse response of the filter with numerator coefficients, b, and denominator coefficients, a. impz chooses the number of samples and returns the response in the column vector, h, and the sample times in the column vector, t. t = [0:n-1]' and n = length(t) is computed automatically.

เช่น

b=[0.1807, 0.1047, 0.3107, 0.1047, 0.1808];

a= [1,-1.1479, 1.5107, -0.6991, 0.2703]

[h,t]=impz(b, a);

จะเห็นว่าเราจะได้ h มีจำนวน 145 ค่า แต่ต้องการจำกัด h ให้มีจำนวนที่ต้องการก็สามารถทำได้ เช่นต้องการเพียงแค่ 50 ค่า เราจะใช้คำสั่งดังนี้

[h,t]=impz(b, a, 50);

จากตารางด้านล่าง จงแปลง IIR เป็น FIR โดย ไม่จำกัดจำนวน h และ จำกัดจำนวน h เพียงแค่ 32 ค่า

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **B & A** | **กราฟ Frequency Response ของ b และ a** | **กราฟ Frequency Response ของ h32** |
| 1 | B=[0.0181 0.0543 0.0543 0.0181]  A=[1.0000 -1.7600 1.1829 -0.2781] |  |  |
| 2 | B=[0.0495 -0.1486 0.1486 -0.0495]  A=[1.0000 1.1619 0.6959 0.1378] |  |  |
| 3 | B=[0.1311 0 -0.2622 0 0.1311]  A=[1.0000 0.4824 0.8101 0.2269 0.2722] |  |  |
| 4 | B=[0.5050 0.3547 1.0723 0.3547 0.5050]  A=[1.0000 0.4824 0.8101 0.2269 0.2722] |  |  |
| 5 | B=[0.5 0.5]  A=[1] |  |  |
| 6 | B=[0.5 -0.5]  A=[1] |  |  |

**คำถาม** ให้นิสิตสังเกต โดยเปรียบเทียบ กราฟ Frequency Response ที่ได้จาก b และ a และจาก h32ต่างกันอย่างไร? และเพราะอะไร?

ตอบ

เหมือนกัน

* **การประมาณองค์ประกอบความถี่ของสัญญาณอินพุต (digital data) ได้จากประยุกต์ใช้ฟังก์ชั่น freqz**

จริงแล้วนิสิตจะได้เรียน Discrete Fourier Transform(DFT) และ Fast Fourier Transform (FFT) ในช่วงท้ายๆของบทเรียนอยู่แล้ว แต่ในการทดลองนี้จะช่วงเสริมความเข้าให้ก่อนในเริ่มต้น

สมมุติ สัญญาณ x[n] ซึ่งเกิดจากผลรวมของสัญญาณ sinusoids 2 ความถี่ คือ 300 Hz ที่ขนาด 5และ 1200 Hz ที่ขนาด 7 โดย sampled ที่ความถี่ 6 kHz. เราอาจหาองค์ประกอบให้วาดกราฟแสดง 100 ลูกคลื่นของสัญญาณ 300 Hz

fs=6000; %% Sampling rate

interval=100\*(1/300); %% Interval time 100 period of 300 Hz

time=0:1/fs:interval;

x= 5\*cos(2\*pi\*300\*time)+7\*cos(2\*pi\*1200\*time); %% Create signal

l=length(x);

[H W]=freqz(x,1,l);

figure;

plot(W\*fs/(2\*pi),abs(H)); %% Plot only magnitude of frequency components

xlabel('Frequency in Hz'); %% X-axis is Frequency in Hz



รูปผลลัพธ์ที่ได้ ซึ่งจะเห็นได้ชัดเจนว่า มีองค์ประกอบความถี่ 300 Hz และ 1200 Hz ที่แรงอย่างชัดเจน

ให้นิสิตอ่านสัญญาณเสียงจากไฟล์ ‘one.wav’ ด้วยคำสั่ง ‘audioread’ แล้วจงประยุกต์ใช้วิธีการข้างบนให้เห็นว่าสัญญาณเสียงนี้มีย่านความถี่อยู่ที่ใดบ้าง

คำสั่ง matlab

def plotSpectrum(y,Fs):

n = len(y) # length of the signal

k = numpy.arange(n)

T = n/Fs

frq = k/T # two sides frequency range

frq = frq[range(n//2)] # one side frequency range

Y = numpy.fft.fft(y)/n # fft computing and normalization

Y = Y[range(n//2)]

plt.plot(frq,abs(Y)) # plotting the spectrum

plt.xlabel('Freq (Hz)')

plt.ylabel('|Y(freq)|')

plt.show()

samp, one = scipy.io.wavfile.read('one.wav')

plotSpectrum(one,samp)

รูปที่ได้และคำตอบ



0- 800 Hz