ID Name .

**Prog #01**

**Introduction Digital Signal Processing**

****

1. **Music Composition**

ในการสร้างเพลงซึ่งประกอบด้วยเครื่องดนตรีหลายชนิดและเสียงร้อง สามารถประมวลทางดิจิตอลนั้นได้โดยนำข้อมูลเสียงดนตรีแบบดิจิตอลชนิดต่างๆมาผสมกัน ด้วยการบวกค่าทางคณิตศาสตร์ ซึ่งในการทดลองนี้เป็นการแสดงให้นิสิตได้เข้าใจกระบวนการสร้างเสียงดนตรีแบบง่าย เพื่อให้ได้เข้าใจในพื้นฐานของการประมวลผลทางดิจิตอล สัญญาณอนาล็อกเป็นสัญญาณที่ต่อเนื่อง ไม่สามารถนำมาประมวลผลได้ในระบบคอมพิวเตอร์ (ส่วนใหญ่ใช้วงจรไฟฟ้า อิเล็กทรอนิคส์ในการจัดการ) สัญญาณอนาล็อกเช่นสัญญาณเสียง สัญญาณอนาล็อกจะผ่านการแปลง การชักค่า (Sampling) และการควอนไทซ์ (quantization) เก็บเข้าเครื่องคอมพิวเตอร์ ข้อมูลที่เก็บในคอมพิวเตอร์ถูกเปลี่ยนเป็นชุดค่าตัวเลขแสดงความสูงของขนาดความดังของเสียง เรียงลำดับตามจังหวะเวลาการชักค่า ในปฏิบัติการครั้งนี้นิสิตจะได้ลงมือประมวลผลข้อมูลเสียง Digital Audio Processing แบบเบื้องต้น นอกจากนั้นนิสิตยังได้ทำการทดลองกับข้อมูลดิจิตอลที่ได้จาก การวัดคลื่นการเต้นของหัวใจ และเขียนโปรแกรมคำนวณหาอัตราการเต้นของหัวใจจากข้อมูลดิจิตอลนั้นด้วย ทั้งเพื่อให้ได้อรรถรสและมุมมองในวิชานี้

**การเตรียมตัวก่อนทดลอง**

เนื่องจากการทดลองนี้จะต้องใช้ Matlab ในการสั่งงาน ดังนั้นนิสิตควรเตรียมศึกษาการใช้งานคำสั่งของ Matlab มาก่อน ความรู้และคำสั่งที่ควรเรียนรู้มากก่อนมีดังนี้

* ข้อมูลแบบเมตริก (Matrix) การสร้าง การอ้างถึงข้อมูลสมาชิกแบบตัวเดียว หรือเป็นกลุ่มข้อมูล การคำนวณผลทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร แต่ละสมาชิกของข้อมูล และฟังชั่นที่ใช้งานกับ matrix เช่น size
* ไฟล์ .wav ที่ใช้เก็บข้อมูลเสียง
* audioread คำสั่งที่ใช้ในการอ่านข้อมูลเสียง
* sound คำสั่งที่นำข้อมูลดิจิตอลออกมาเป็นเสียงอนาล็อก
* plot คำสั่งที่ใช้วาดกราฟเส้นจากชุดลำดับข้อมูล
* load ที่ใช้อ่านข้อมูลจากไฟล์ที่เก็บข้อความไว้

1. **การทดลอง**

2.1 สมมุติเรามีการบันทึกเสียงดนตรีและได้แปลงมาเป็นไฟล์ข้อมูลดิจิตอลแล้ว ประกอบไปด้วย ไฟล์ bass.wav guitar.wav และ drums.wav

ให้นิสิต อ่านข้อมูลเสียงทั้งหมด ทดลองเล่นเสียง และ plot ภาพสัญญาณดนตรีทั้งสามลงบน figure เดียวกัน

นำ code ใส่พื้นที่นี้

>> [guitar, fs] = audioread(‘guitar.wav’);

>> [bass, fs] = audioread(‘bass.wav’);

>> [drums, fs] = audioread(‘drums.wav’);

>> sound(guitar, fs);

>> sound(bass, fs);

>> sound(drums, fs);

>> subplot(3,1,1);

>> plot(guitar);

>> xlabel('Time(second)');

>> ylabel('Amplitude');

>> title('Guitar');

>> subplot(3,1,2);

>> plot(bass);

>> xlabel('Time(second)');

>> ylabel('Amplitude');

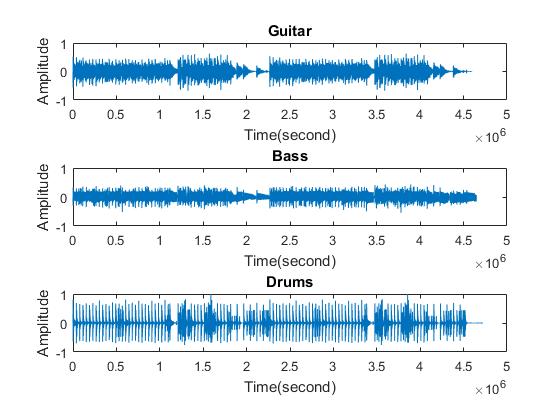
>> title('Bass');

>> plot(drums);

>> xlabel('Time(second)');

>> ylabel('Amplitude');

>> title('Drums');

**รูป

*ความถี่ของการสุ่มสัญญาณเสียงของทั้งสามดนตรี* = 44100 samples/sec

*ความยาวของเสียงดนตรี* bass =105.3671 วินาที guitar = 104.1516 วินาที และ drums =107.1650 วินาที

*ลักษณะการเก็บข้อมูลเสียงดนตรีเป็นแบบใด (ใช้คำสั่ง size)* รูปแบบ Array.

2.2 ให้นิสิตตัดเสียงทั้งสามดนตรีมาอย่างละ 20 วินาที เริ่มตัดที่วินาทีที่ 10 แล้วทำการผสมเสียงดนตรีเข้าด้วยกันโดยนำค่าข้อมูลแต่ละค่าจากสามเสียงดนตรีมาบวกกัน แล้วเก็บผลลัพธ์ไว้ในตัวแปรชื่อ song จากนั้นให้ทดสอบฟังเสียงดู และทำการ plot สัญญาณเสียงเพลงที่รวมได้

นำ code ใส่พื้นที่นี้

>> [guitar, fs] = audioread(‘guitar.wav’);

>> [bass, fs] = audioread(‘bass.wav’);

>> [drums, fs] = audioread(‘drums.wav’);

>> guitar\_seg = guitar(fs\*10 : fs\*30);

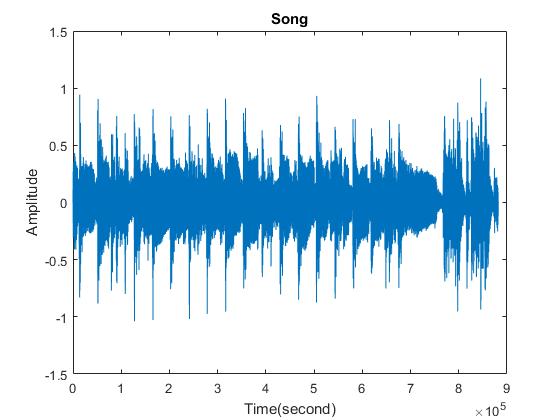
>> bass\_seg = bass(fs\*10 : fs\*30);

>> drums\_seg = drums(fs\*10 : fs\*30);

>> song = guitar\_seg + bass\_seg + drums\_seg;

>> sound(song, fs);

>> plot(song);

**รูป

*เสียงเพลงผสมได้เป็นอย่างไร (อธิบายเชิงพรรณนา)*

*เสียงที่ได้ยิน เป็นเสียงที่ได้ยินทุกเสียงที่มีความดังเท่ากัน หรือก็คือได้ยินเสียงทุกชนิด ที่เสมือนเล่นอยู่ในตำแหน่งเดียวกัน*

*รูปภาพที่ plot เป็นอย่างไรเมื่อเปรียบกับเสียงดนตรีทั้งสามที่แยกกัน*

*เหมือนเป็นเสียงเนื้อเดียวที่เหมือนกับเสียงดนตรีต้นฉบับสามเสียง ดูไม่เหมือนการเอาสามเสียงมารวกันเลย*

*ถ้าหากเสียงดนตรีทั้งสามถูกสุ่มเก็บ(ขั้นตอนการแปลงข้อมูลดิจิตอล)ด้วยอัตราสุ่มที่ไม่เท่ากัน นิสิตคิดว่ายังคงสามารถใช้วิธีการข้างต้นในการผสมเพลงได้อีกหรือไม่ เพราะอะไร*

*อาจจะทำได้ โดยไฟล์เอาท์พุท จะมีความถี่ตามเสียงที่มี sampling rate น้อยสุด*

1.3 ให้เพิ่มขนาดความดังของเสียงกีต้าร์จาก 0 จนถึง 3 เท่าของความดังเดิมตามระยะเวลา (หมายถึงที่วินาทีแรกๆเสียงเบา จนถึงช่วงปลาย ๆ เสียงดังเป็น 3 เท่า ดังเพิ่มขึ้นแบบเชิงเส้น) plot ดูรูปสัญญาณ แล้วจึงให้ผสมสัญญาณเสียงเพลงใหม่ โดยใช้เสียงกีต้าร์ที่ถูกปรับเสียงในการผสมเพลง ทดลองฟังเสียงเพลงที่ผสมใหม่พร้อมสังเกตเสียงที่ได้

>> [guitar, fs] = audioread(‘guitar.wav’);

>> [bass, fs] = audioread(‘bass.wav’);

>> [drums, fs] = audioread(‘drums.wav’);

>> guitar\_seg = guitar(fs\*10 : fs\*30);

>> linear = linspace(0.0, 3.0, length(guitar\_seg));

>> linear = transpose(linear);

>> guitar\_linear = guitar\_seg .\* linear;

>> bass\_seg = bass(fs\*10 : fs\*30);

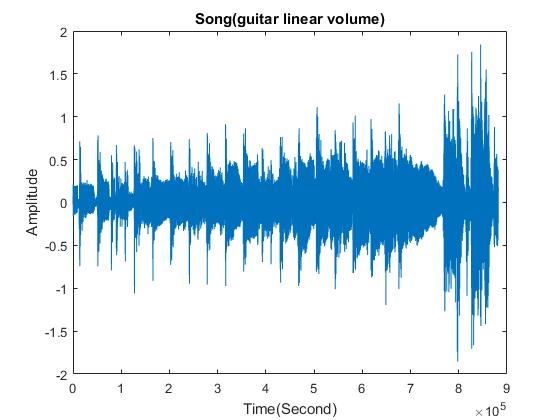
>> drums\_seg = drums(fs\*10 : fs\*30);

>> song = guitar\_linear + bass\_seg + drums\_seg;

>> sound(song, fs);

>> plot(song);

รูป

**

1.4 เสียงเพลงข้างต้นเป็นเพลงแบบ mono ซึ่งจะเป็นเสียงที่ถูกบันทึกช่องสัญญาณเดียว ดังนั้นเวลาเล่นกลับออกมาก็มีสัญญาณช่องเดียวเหมือนกันไม่ว่าจะมีกี่ลำโพงก็ตามเสียงจะออกมาเหมือนๆ กัน ส่วนเพลง Stereo จะเป็นเสียงถูกบันทึกสองช่องสัญญาณ การเล่นเพลงแบบนี้เสียงที่ออกมาจะไม่เหมือนกันทำให้สามารถจำลองสถานการณ์เสียงแบบซ้าย-กลาง-ขวาได้ครับเช่น อาจให้เสียงคนร้องอยู่ตรงกลาง กีตาร์เยื้องไปทางขวาเล็กน้อยส่วนกลองอยู่ด้านซ้ายหน่อยๆ เสียงประสานอยู่ขวาสุด เป็นต้น ตัวอย่างต่อไปนี้เป็นเสียงเพลง Stereo คือเพลง ‘road.wav’ และ ‘hootie.wav’ ให้นิสิตอ่านข้อมูลเพลงทั้งสองใน Matlab แล้วทำการแยกข้อมูลช่องซ้ายขวาออกจากกัน จากนั้นสร้างเพลงใหม่ที่เกิดจากสัญญาณช่องซ้ายหักด้วยช่องขวา ตามนี้

[road,fs]=audioread('road.wav');

sound(road,fs);

left=road(:,1);

right=road(:,2);

sound(left-right,fs);

*เสียงเพลงทั้งสองต่างกันอย่างไร*

*ในการเล่นครั้งแรก เสียงร้องกับเสียงดนตรีอยู่คนละทิศทางกัน ทำให้ฟังเนื้อร้องรู้เรื่อง*

*แต่ในการเล่นครั้งที่สอง เสียงร้องกับดนตรีทับกัน เหมือนอยู่ตำแหน่งเดียวกันทำให้ฟังไม่รู้เรื่อง*

*เหตุใดจึงเป็นเช่นนั้น*

*เพราะการเล่นครั้งที่สองเหมือนเป็นการนำเสียงมาเล่นในตำแหน่งเดียวกัน และไม่สามารถแยกเสียงได้ รวมกันเป็นเสียงเนื้อเดียว mono*

ให้ทดลองทำซ้ำเช่นเดียวกันข้างต้นกับ เพลง ‘hootie.wav’ แล้วทดลองฟังเสียงเพลงผลลัพธ์

*เสียงเพลงเกิดจากสัญญาณช่องซ้ายหักด้วยช่องขวา ยังมีเสียงคนร้องอยู่หรือไม่*  ยังคงมีอยู่แต่ไม่ชัด

*เหตุใดจึงเป็นเช่นนั้น*

*เช่นเดียวกับกรณี road.wav เนื้อเสียงเป็นเสียงเดียว เล่นในตำแหน่งเดียวกัน ทำให้ได้ยินเสียงเป็น mono*

1. ไฟล์ข้อมูล “ecg.txt” คือข้อมูลดิจิตอลของสัญญาณไฟฟ้าที่ใช้ตรวจจับอัตราการเต้นของหัวใจมนุษย์ สัญญาณดังกล่าวถูกสุ่ม (sampling) ที่ 100 samples/sec นิสิตจงทำตามคำสั่ง ต่อไปนี้

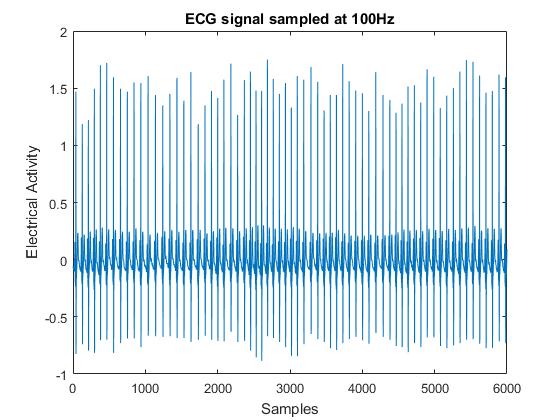
sig=load('ecg.txt');

plot(sig);

xlabel('Samples');

ylabel('Electrical Activity');

title('ECG signal sampled at 100Hz');

**รูป

นิสิตจงเขียนโปรแกรม matlab คำนวณหาอัตราการเต้นของหัวใจจากข้อมูลในไฟล์ “ecg.txt”

*นิยามให้ได้ว่าส่วนใดในรูปคือการเต้นของหัวใจ*

*สังเกตจุดยอดของคลื่นดังกล่าว เราสามารถนับได้ว่าเมื่อคลื่นกำลังขึ้นถึงจุดยอดนั้นเป็นการเต้นหนึ่งจังหวะของหัวใจ*

*แนวคิดในการคำนวณหา (นิสิตต้องคิดว่าในโปรแกรมนั้นจะเห็นเพียงข้อมูลตัวเลขตามลำดับ ใช้นิยามข้างแล้วอธิให้เข้าใจแนวคิดการค้นหาจังหวะการเต้นของหัวใจ)*

แนวคิด

*ใช้วิธีการนับจุดยอดของกราฟก็จะได้อัตราการเต้นของหัวใจ จากกราฟแสดงผลถึง 6000 samples จากข้อมูลเป็นการ sampling ที่ 100 sample/second ดังนั้นเมื่อนับจุดยอดในกราฟแล้วจะได้อัตราการเต้นของหัวใจ ต่อ หนึ่งนาที พอดี*

นำ code ใส่พื้นที่นี้

sig=load('C:\Users\SKpoTH\Desktop\Signal Transformation Technic\work\work1\ecg.txt');

plot(sig);

xlabel('Samples');

ylabel('Electrical Activity');

title('ECG signal sampled at 100Hz');

count\_peak = 0;

found\_peak = 0;

for i = 1:length(sig)

if (found\_peak == 0)

if (sig(i) > 0.5)

count\_peak = count\_peak+1;

found\_peak = 1;

end

elseif (found\_peak == 1)

if (sig(i) < 0.5)

found\_peak = 0;

end

end

end

%ดูค่าที่ตัวแปล count\_peak จะได้อัตราการเต้นของหัวใจต่อหนึ่งนาที

*อัตราการเต้นของหัวใจ* 67 ครั้ง/นาที