

การสร้างเสียงอิเล็กทรอนิกส์สำหรับกีตาร์โดยคอมพิวเตอร์

Computer Guitar Effect

นายเกรียงไกร เสียงกล่อม 49211506 นางสาว ภูยาภรณ์ เจียรนันทนา 49211513
 นายนภัสสุพันธ์ นาคพงษ์ 49211522 นายปรพัฒน์ เพชรคำรังค์สกุล 49211525 นายสิทธิชัย ไชยพร 49211547
 คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ชั้นปีที่ 2 ภาคเรียนที่ 2 ปีการศึกษา 2550
 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อศึกษาหลักการประมวลผลสัญญาณเสียง โดยนำความรู้ที่ได้เรียนจากวิชา CPE222 Signals and Systems มาประยุกต์ใช้กับชีวิตจริง เสียงที่นำมาประมวลผลสัญญาณนี้เป็นเสียงที่ได้จากกีตาร์ไฟฟ้า อัดโดยโปรแกรม Sound recorder แล้วนำมาใส่ Effect โดยโปรแกรม MATLAB ซึ่ง Effect ที่ค้นคว้าก็คือ Single Echo และ Multiple Echo ซึ่งเป็น Effect พื้นฐานในการประมวลผลสัญญาณ ทางกลุ่มสามารถสร้าง Effect และทำการหา Impulse response ของระบบได้ ซึ่งรายละเอียดอยู่ในเนื้อหาต่อไปนี้

1. บทนำ

วิชา Signal and System เป็นวิชาที่เกี่ยวข้องกับการประมวลผลสัญญาณ รูปแบบของสัญญาณที่ใกล้ตัวในชีวิตประจำวันมากที่สุดและง่ายต่อการประมวลผลมากที่สุด หนึ่งไม่พ้นเรื่องภาพและเสียง เนื่องจากสมาชิกในกลุ่มนี้ชอบการเล่น กีตาร์อยู่แล้ว เลยก็คิดว่าจะทำโครงการกับลิ่งที่ชื่นชอบนี้ได้ดี กีตาร์ที่ใช้เล่นก็ไม่พ้นกีตาร์ราคาถูกทั่วไปทั้งกีตาร์โปร่งและกีตาร์ไฟฟ้า กีตาร์ไฟฟ้านั้น มีข้อดีกวากีตาร์โปร่งตรงที่สามารถนำสัญญาณเสียงที่เล่นได้เข้าสู่เครื่องสร้าง Effect และ Amplifier ต่อเข้าสู่ลำโพงเพื่อเล่นออกสู่ผู้ฟัง Effect ที่ว่านี้ต้องซื้อเป็นอุปกรณ์สำหรับปูมจากร้านขายอุปกรณ์การเล่นดนตรี ทำให้เสียงที่ได้มีจำกัดแล้วแต่คุณภาพของเครื่อง Effect นั้น เช่นถ้า Effect นั้นราคาถูก ประมาณ 1000 บาทก็จะเล่น Effect แปลกดๆ ได้ไม่

มาก เช่น การ Distortion เป็นการทำให้เสียงกีตาร์นั้นขาดและแตกพร่า เป็นต้น นิยมใช้มากในการเล่นเพลงแนว String , Rock เป็นต้น ถ้าหากได้ Effect ที่ดีแบบอื่นๆ ก็ต้องเสียเงินซื้อ ที่เคยเห็นก็ราคาประมาณ 3,000 บาทขึ้นไป ก็สามารถปรับแต่ง Effect ได้ ประมาณมากกว่า 50 - 100 ชนิด ถ้าหากเอานแบบมืออาชีพราคาคงหนีไม่พ้นหลักหมื่นอย่างแน่นอน

พอเราเรียนวิชา Signals and Systems ก็รู้ว่าเสียงสามารถนำเข้าสู่คอมพิวเตอร์เพื่อประมวลผลได้ ส่ง Output ออกทางลำโพง ก็เลยไปค้นคว้าหาข้อมูลที่ www.musicdsp.org และ harmony-central.com เป็นเว็บที่ว่าด้วยแนวทางการทำ Guitar Effect ด้วยคอมพิวเตอร์ ก็เลยอยากรู้ว่า Effect ที่สร้างเป็นแบบเฉพาะของตนเองขึ้นมาซักชิ้น อย่างน้อยก็ไม่ซ้ำแบบใคร อีกทั้งราคาถูกปรับแต่งได้ตามใจ และได้ใช้ความรู้เรื่อง Signals and Systems อย่างเต็มที่ในการทำโครงการนี้อีกด้วย

2. ขอบเขตของโครงการ

- ใช้ได้กับกีตาร์ไฟฟ้า หรือ กีตาร์โปร่งเท่านั้น ต้องนำเสียงจาก กีตาร์เข้า Computer ประมาณผลแล้วส่งออกทางลำโพง

- มี Effect ที่ทำได้มี 2 ชนิดคือ Single Echo และ Multiple Echo โดยแต่ละชนิดสามารถปรับแต่งได้เพิ่มเติมตามต้องการ

- ถ้าต้องการให้เสียงที่ได้ไฟเราจะใช้อุปกรณ์ที่เรียกว่า Pick up อัดเสียงจากกีตาร์โปร่งเข้า

คอมพิวเตอร์ หรือ เครื่องคอมพิวเตอร์ต่อเข้าคอมพิวเตอร์ได้โดยตรงเลย

4. เครื่องคอมพิวเตอร์นั้นต้องมีโปรแกรม MATLAB Version 6.5 ขึ้นไป

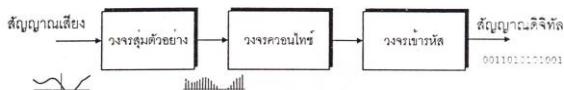
5. ปรับแต่ง Effect โดยปรับค่าตัวเลขใน Code โปรแกรม MATLAB เท่านั้น

3. การดำเนินงาน

3.1 ทฤษฎีเกี่ยวกับข้อง

3.1.1 การประมวลสัญญาณเสียงด้วย MATLAB

สัญญาณเสียง จัดว่าเป็นสัญญาณแอนalog (Analog) ประเภทหนึ่งที่มีค่าไม่แน่นอน เป็นลักษณะไปตามเวลา ดังนั้นในการส่งสัญญาณเสียงผ่านช่องสัญญาณสื่อดิจิทัล (Digital) จึงจำเป็นที่ภาคส่งจะต้องนำสัญญาณเสียงไปผ่านกระบวนการสุ่มตัวอย่างก่อนเพื่อให้ได้เป็นสัญญาณที่มีค่า Discrete ทางเวลา จากนั้นจึงนำสัญญาณ Discrete ที่ได้แต่ละแซมเปิด (Sample) ไปผ่านกระบวนการ量化 (quantization) เพื่อให้ได้สัญญาณที่มีค่า Discrete ทางแอมพลิจูด (Amplitude) ก่อนจะนำไปเข้ารหัส (binary encoding) เป็นตัวเลขในนรี 0 และ 1 เป็นลำดับต่อไป แผนภาพโดยรวมของวงจรแปลงสัญญาณเสียงเป็นสัญญาณดิจิทัลเป็นดังนี้



3.1.1.1 การอ่านและบันทึกไฟล์เสียงด้วย MATLAB

ในโปรแกรม MATLAB มีคำสั่ง wavread ที่ใช้สำหรับอ่านไฟล์เสียง (.wav) เข้ามาเก็บไว้ในตัวแปร เวกเตอร์ที่กำหนดเพื่อนำไปประมวลผลต่อไป รูปแบบการใช้งานคำสั่งมีลักษณะดังนี้คือ

>> y = wavread(FILE);

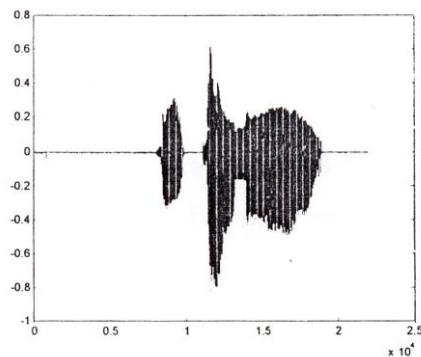
ยกตัวอย่างเช่นเรามีไฟล์เสียงชื่อ voice_sample.wav อยู่ในharvardic แล้วต้องการอ่านไฟล์ดังกล่าว ให้พิมพ์คำสั่งดังนี้

>> y = wavread('voice_sample.wav');

ค่าแอมพลิจูดของสัญญาณเสียงในไฟล์จะได้รับการบรรจุลงในเวกเตอร์ y โดยที่ค่าแอมพลิจูดเหล่านี้จะมีขนาด [-1,+1] ในที่นี้ไฟล์ดังกล่าวบรรจุเสียงของคำว่า “ดอกกล้อยไม้” เราสามารถดูรูปสัญญาณเสียงที่อยู่ในไฟล์ดังกล่าวโดยใช้คำสั่งดังนี้

>> plot(y);

ผลลัพธ์ที่ได้จะมีลักษณะดังนี้



เราสามารถสั่งให้เครื่องคอมพิวเตอร์อ่านไฟล์เสียงและเล่นเสียงออกทางลำโพงได้โดยคำสั่ง

>> wavplay(y);

ในการเล่นเสียงนี้เราสามารถกำหนดความเร็วในการเล่นได้ตามต้องการ คือจะเล่นให้เร็วขึ้นกว่าปกติหรือช้าลงก็ได้ แต่ก่อนอื่นเราจำเป็นต้องทราบถึงอัตราการสุ่มตัวอย่าง (sampling rate) ของไฟล์เสียงดังกล่าวก่อน ค่าอัตราการสุ่มถูกกำหนดในขั้นตอนการอัดเสียง ซึ่งโดยทั่วไปการใช้อัตราการสุ่มเพิ่มสูงมากขึ้นเท่าไหร่ กุณภาพของสัญญาณเสียงที่ได้มักจะดีขึ้นและใกล้เคียงกับสัญญาณเสียงดั้นฉบับมากขึ้น อย่างไรก็ตามขนาด

ของไฟล์ที่บรรจุเสียงก็จะต้องมีขนาดใหญ่ขึ้นด้วย ค่าอัตราการสุ่มตัวอย่างสามารถอ่านได้จากไฟล์เสียงโดยใช้คำสั่งต่อไปนี้

```
>> [y fs] = wavread('voice_sample.wav');
```

ผลที่ได้คือตัวแปร y จะบรรจุแอมพลิจูดของสัญญาณเสียง ในขณะที่ตัวแปร fs จะบอกถึงอัตราการสุ่มของสัญญาณ สำหรับไฟล์ตัวอย่างที่ใช้มีค่าอัตราการสุ่มเท่ากับ $F_s = 11,025$ ครั้งต่อวินาที จากค่าอัตราสุ่มที่ได้จะนำมาใช้ในการควบคุมความเร็วของการเล่นเสียงกลับได้ดังนี้

```
>> wavplay(y,fs*2);
```

คำสั่งนี้เป็นการเร่งให้เครื่องคอมพิวเตอร์เล่นเสียงเร็วขึ้นกว่าปกติ 2 เท่า แต่หากต้องการให้เครื่องคอมพิวเตอร์เล่นเสียงช้าลงให้ใช้คำสั่งต่อไปนี้

```
>> wavplay(y,fs*0.5);
```

พิจารณารูปสัญญาณเสียงในรูปข้างต้นอีกครั้ง จะเห็นว่าในแกน x มีแซมเพลเสียงหรือจำนวนจุดอยู่ทั้งหมด 25,000 จุด เนื่องจากอัตราการสุ่มนี้มีค่าเท่ากับ $11,025$ ครั้งต่อวินาทีแสดงว่า สัญญาณเสียงที่บันทึกอยู่ในไฟล์มีระยะเวลาเท่ากับ $25,000/11,025 = 2.27$ วินาที ในกรณีที่เราเร่งความเร็วของการเล่นเสียงกลับขึ้นเท่าตัวระยะเวลาของเสียงที่ออกจากลำโพงก็จะสั่นลงครั้งหนึ่งคือเหลือเพียง 1.13 วินาที ในทางกลับกันหากมีการลดความเร็วของการเล่นกลับระยะเวลาของเสียงที่ได้ก็จะนานขึ้นตามสัดส่วนที่กำหนด

ในการเล่นเสียงกลับผู้ใช้สามารถกำหนดให้โปรแกรม MATLAB เลือกเล่นเฉพาะบางช่วงของสัญญาณได้ ยกตัวอย่าง เช่น เราต้องการเล่นกลับสัญญาณเสียงเฉพาะพยางค์แรก ให้ใช้คำสั่งดังนี้

```
>> wavplay(y(5000:10000));
```

ผลลัพธ์ที่ได้คือเครื่องคอมพิวเตอร์จะให้คำนิยมเสียงเฉพาะคำว่า “ดอก” ในกรณีที่จะเล่นเสียงเฉพาะคำว่า “กลวย” หรือ “ไม้” ก็สามารถทำได้ในลักษณะเดียวกันอย่างไรก็ตามสังเกตว่ารูปสัญญาณของคำว่า “กลวย” และ “ไม้” ค่อนข้างจะติดกันมาก ต่างจากในกรณีของคำว่า “ดอก” ที่รูปสัญญาณแยกออกจากพยางค์ที่เหลืออย่างชัดเจน

โปรแกรม MATLAB สามารถที่จะตัดเสียงได้ตามที่เราต้องการและสามารถที่จะนำเสียงที่ตัดต่อได้จัดเก็บเป็นไฟล์ในคอมพิวเตอร์ได้อีกด้วย จากเนื้อหาที่ผ่านมาทำให้เราทราบว่าการกำหนดช่วงของตัวแปร y จะทำให้เราเล่นเสียงเป็นช่วงได้ เราสามารถนำช่วงที่ได้เก็บในตัวแปรอื่น และจัดเก็บเป็นไฟล์ได้ โดยใช้คำสั่งดังนี้

```
>> x = y(500:10000);
>> wavwrite(x,fs,'voice_sample2.wav');
```

ไฟล์ voice_sample2.wav จะเป็นเสียงของคำว่า “ดอก”

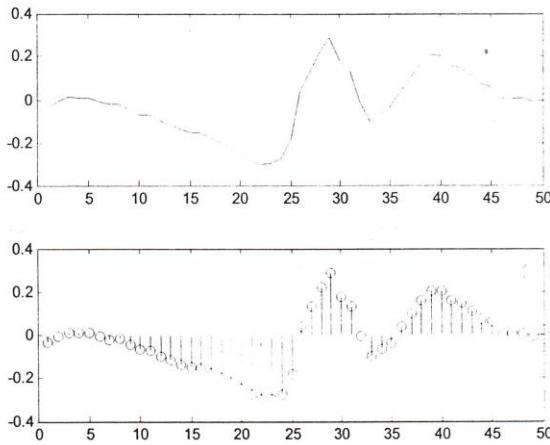
3.1.1.2 การทำซับแท่มปลิง

ก่อนอื่นให้อ่านไฟล์สัญญาณเสียง

voice_sample2.wav ที่บรรจุคำว่า “ดอก” อูໝມาก็เป็นในตัวแปร x จากนั้นพิจารณาสัญญาณเสียงดังกล่าวช่วงสั้นๆ เช่น เพียง 50 แซมเพลโดยใช้คำสั่งต่อไปนี้

```
>> [x,fs] = wavread('voice_sample2');
>> subplot(2,1,1);
>> plot(x(4000:4049));
>> subplot(2,1,2);
>> stem(x(4000:4049));
```

จะได้ผลดังรูป



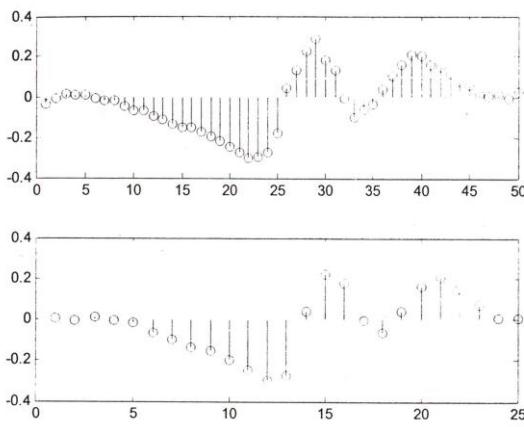
สังเกตว่ารูปสัญญาณที่ได้จากการคำสั่ง plot จะมีลักษณะต่อเนื่อง ในขณะที่รูปสัญญาณที่ได้จากการคำสั่ง stem จะให้ผลการแสดงในรูปของสัญญาณ Discrete ในลำดับต่อไปนี้จะนำสัญญาณเสียงในตัวแปร x ไปผ่านกระบวนการซับแซมป์лин โดยรายละเอียดจำนวนแซมเปิลลงไปครึ่งหนึ่ง โดยคำสั่งต่อไปนี้

```
>> L = length(x);
>> n = 1:L;
>> y = x(n);
```

นำสัญญาณเสียงต้นฉบับในตัวแปร x ไปเทียบกับสัญญาณเสียง y โดยอาศัยคำสั่งต่อไปนี้

```
>> subplot(2,1,1);
>> stem(x(4000:4049));
>> subplot(2,1,2);
>> stem(y(2000:2024));
```

ผลลัพธ์ที่ได้เป็นดังนี้คือ



สังเกตกระบวนการซับแซมป์линช่วยให้เราลดจำนวนหน่วยความจำในการเก็บสัญญาณลงไปครึ่งหนึ่ง โดยรูปสัญญาณข้าง Kong มีลักษณะใกล้เคียงสัญญาณต้นฉบับ เราสามารถทดสอบคุณภาพของสัญญาณเสียงที่ได้โดยใช้คำสั่ง wavplay(y,fs/2) และพบว่าเสียงเพี้ยนไปจากเดิมเล็กน้อย หากเราทำซ้ำกระบวนการซับแซมป์лин กับสัญญาณใหม่นี้อีกครั้ง สัญญาณเสียงที่ได้จากการทำซับแซมป์линครั้งใหม่นี้จะมีรูปสัญญาณที่ยังแตกต่างจากรูปสัญญาณต้นฉบับมากขึ้น หากเราลดจำนวนแซมเปิลต่อหนึ่งหน่วยเวลาของสัญญาณเสียงลงด้วยการทำซ้ำกระบวนการซับแซมป์лин ผลที่ได้คือในที่สุดรูปสัญญาณที่ได้จะไม่เหลือเก้าโครงเดิมของสัญญาณต้นฉบับอีกเลย ถ้าเราทดลองฟังเสียงจะพบว่าเสียงที่ได้ผิดเพี้ยนไปจากเสียงต้นฉบับมากจนไม่สามารถฟังได้ว่าเป็นคำพูดว่าอะไร

จากการทดลองนี้ทำให้เราเข้าใจถึงการทำงานส่วนของสัญญาณเสียงนั้นมีความเกี่ยวข้องกับคุณภาพของเสียงที่ได้และขนาดของไฟล์สัญญาณเสียงด้วย ถ้าอัตราการสูงมาก เสียงจะมีความใกล้เคียงกับเสียงต้นฉบับมากขึ้น ถ้าอัตราการสูงน้อยเสียงจะมีลักษณะที่ใกล้เคียงกับสัญญาณเสียงต้นฉบับน้อยลง

3.1.2. การประมวลสัญญาณเสียงดนตรี (Musical Sound Processing)

วิธีการประมวลสัญญาณเสียงดนตรีมีวิธีหลักๆ 2 วิธี ในขั้นแรกเป็นการนำเข้าเสียงของเครื่องดนตรีแต่ละชนิดซึ่งยังไม่ได้ผ่านกระบวนการขยายเสียงเข้าสู่คอมพิวเตอร์ ในโครงการนี้ก็คือการนำเสียงจากกีตาร์ไฟฟ้าเข้าสู่คอมพิวเตอร์ผ่านทางช่อง input ของการคัดเสียง จากนั้นเป็นการจัดการเสียงที่นำเข้ามาด้วยการเพิ่ม Effect ให้กับเสียงนั้นๆ และนำเสียงต้นฉบับเดิมกับเสียงที่ได้เพิ่ม Effect แล้วรวมกัน เสียงที่สร้างขึ้มนั้นเกิดจากหลายสัญญาณ และสามารถเพิ่มสัญญาณต่างๆ ได้โดยใช้เทคนิคการประมวลผลสัญญาณ (Signal Processing) ซึ่งเทคนิคต่างๆ อยู่ในเนื้อหาถัดไปนี้

3.1.2.1 Time-Domain Operations

ตัวนวัตกรรมจะใช้ Time-domain operation ในการสร้างเสียง Effect ให้กับเสียงของคนตี เช่น เสียงแบบ flanging, chorus, single echo, multiple echo และในแต่ละเสียง Effect ต้องใช้ฟื้นฐานเกี่ยวกับการ delay เพราะฉะนั้นการประมวลผลสัญญาณเสียงในที่นี้มักจะอยู่ในรูป Difference equation ซึ่งเป็นรูปแบบสมการที่อยู่ใน Time Domain

3.1.2.2 Single Echo Filter

เสียง Echo นี้เป็นเสียง Effect พื้นฐานแรก สุดที่ควรรู้ โดยสร้างขึ้นมาจากการนำเสียงต้นฉบับมาร่วมกับ Effect แบบ Single echo คือเป็นเสียงที่ผ่านการ Delay แล้วสะท้อนกลับมา จินตนาการได้จากการที่เสียงก้องของเรา วิ่งไปกระแทกกำแพง เสียงที่สะท้อนกลับจะดังน้อยลงและเป็นเสียงที่ผ่านไปช่วงเวลาหนึ่ง โดยที่ sampling periods R จะถูกสร้างขึ้นโดย filter ที่ชื่อว่า FIR filter โดยดูจาก Block Diagram รูปที่ 1.1 (a) และนำมาเขียนสมการ Difference equation ได้ดังนี้

$$y[n] = x[n] + \alpha x[n - R], \quad |\alpha| < 1 \quad (1.1)$$

ซึ่งสามารถหา Transfer function ได้ดังนี้

$$H(z) = 1 + \alpha z^{-R} \quad (1.2)$$

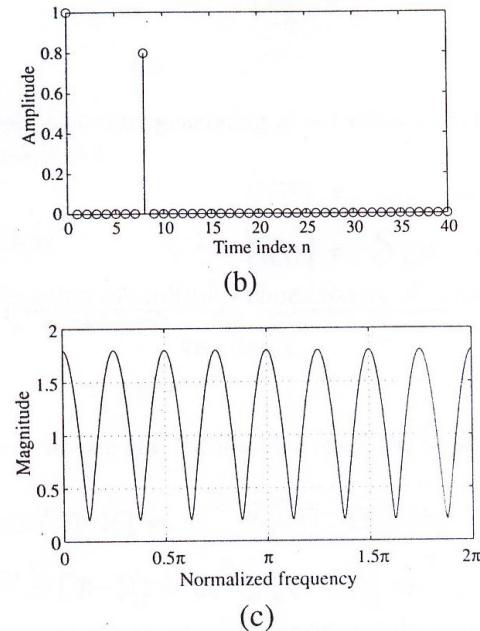
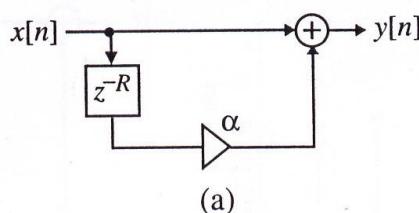


Figure 1.1 Single echo filter: (a) filter structure, (b) typical impulse response, and (c) magnitude response for $R = 8$ and $\alpha = 0.8$

จากในสมการที่ 1.1 และสมการที่ 1.2 มี R เป็นตัวแปรเพื่อบอกค่าการ Delay เมื่อคลื่นเสียงถูกส่งจากแหล่งกำเนิดเสียงไปทางผู้ฟังหลังจากการที่คลื่นเสียงนั้นไปชนกับผนังกันเสียงแล้วสะท้อน ในขณะที่ตัวแปร α ต้องมีค่า $|\alpha| < 1$ และแสดงให้เห็นว่าสัญญาณจะมีค่าน้อยลงเนื่องจากสัญญาณนั้นถูกขยายไปรอบทิศทาง แล้วสะท้อนกลับ

ค่า Impulse response สำหรับสัญญาณนี้อธิบายด้วยรูป 1.1 (b) ซึ่งค่า Magnitude response เสียง Effect แบบ Single echo ซึ่งค่า $\alpha = 0.8$ และค่า $R = 8$

การหา impulse response ของระบบ Single echo filter

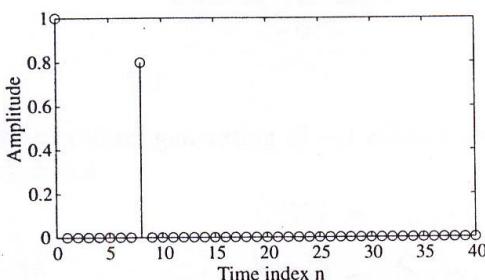
$$\begin{aligned} \text{จาก } \quad y[n] &= x[n] + \alpha x[n - R] \\ \text{ให้ } \quad x[n] &= \delta[n] \end{aligned}$$

$$\begin{array}{ccc} \delta[n] & \longrightarrow & h[n] \\ & \boxed{\quad} & \\ h[n] & = & \delta[n] + \alpha \delta[n - R] \end{array}$$

ให้ $\alpha = 0.8$, $R = 8$ จะได้

$$h[n] = \delta[n] + 0.8\delta[n - 8]$$

ตามรูป 1.1 (d)



รูป 1.1 (d) แสดง impulse response ของระบบ

วิธีการพิสูจน์เพื่อหา Transfer function ของระบบ

$$\text{จาก } y[n] = x[n] + \alpha x[n - R]$$

$$\begin{aligned} x[n] &\xleftrightarrow{Zu} X(z) \\ ax[n - R] &\xleftrightarrow{Zu} z^{-R} X(z) ; \text{ Time Shift property} \\ y[n] &\xleftrightarrow{Zu} Y(z) = X(z) + az^{-R} X(z) \end{aligned}$$

$$Y(z) = X(z) (1 + az^{-R})$$

$$\begin{aligned} H(z) &= Y(z) / X(z) \\ &= 1 + az^{-R} \end{aligned}$$

$$H(z) = 1 + az^{-R}$$

3.1.2.3 Multiple Echo Filter

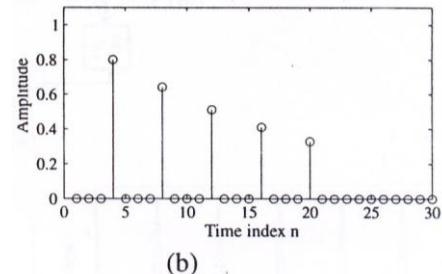
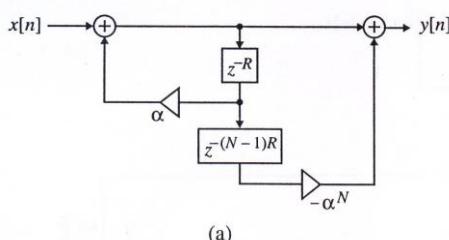


Figure 1.2 Multiple echo filter generating N-1 echoes: (a) filter structure and (b) impulse response with $\alpha = 0.8$ for $N = 6$ and $R = 4$.

การทำสัญญาณเดียงให้มี Effect แบบ Multiple echo ซึ่งเป็นเดียง echo ที่ดังติดตามต่อเนื่องกัน โดยมี ระยะ R ที่ห่างกันและ amplitude ของสัญญาณจะ ลดลงแบบ exponential และใช้ FIR filter ที่อยู่ในรูป transfer function นี้

$$\begin{aligned} H(z) &= 1 + az^{-R} + 2az^{-2R} + 3az^{-3R} + \dots \\ &\quad + (N-1)az^{-(N-1)R} \end{aligned}$$

โดยสามารถ filter ได้จากรูป 1.2 (a) และ impulse response ของ Multiple echo filter มีค่า $\alpha = 0.8$, $N = 6$ และ $R = 4$ โดยดูจากรูป 1.2 (b)
ถ้าเป็นเลขจำนวนอนันต์ของอพอเฟกต์แบบนี้ จะลดลงแบบ exponential ทำให้สู้เช่นกัน หนึ่ง สามารรถหา Transfer Function ใหม่ได้ดังนี้

$$H(z) = 1 + az^{-R} + 2az^{-2R} + 3az^{-3R} + \dots$$

$$H(z) = 1 / 1 - az^{-R} ; |a| < 1$$

ถ้าต้องการสมการในรูป Difference equation ก็สามารถแปลง z-transform กับ และได้ impulse response ดังนี้

$$h[n] = \delta[n] + a\delta[n-R] + 2a\delta[n-2R] + 3a\delta[n-3R] + \dots + Na\delta[n-(N-1)R]$$

หรือสมการ Difference equation นี้นั่นเอง

$$\begin{aligned} y[n] &= x[n] + a[x[n-R] + 2ax[n-2R] \\ &\quad + 3ax[n-3R] + \dots + Nas[n-(N-1)R]] \end{aligned}$$

3.2 วิธีการทำ Effect Guitar ด้วยโปรแกรม MATLAB

1. นำเข้าไฟล์เสียงสู่คอมพิวเตอร์

- โดยต่อ Jack กีตาร์ไฟฟ้าเข้าทางช่อง input ของ sound card และอัดเสียงโดยโปรแกรม sound recorder

2. นำไฟล์เสียงเข้าสู่โปรแกรม MATLAB

- โดยการกู้ปี้จากต้นฉบับไปไว้ที่ C:\MATLAB6p5\work

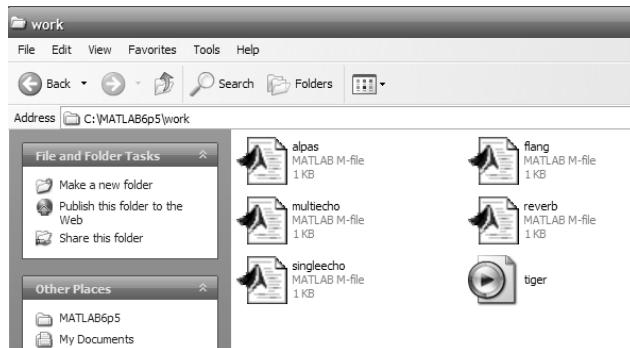


Fig. 2 กู้ปี้ไฟล์ต้นฉบับมาไว้ที่ C:\MATLAB6p5\work

3. การ import ไฟล์เสียง

- เลือกที่ Current Directory

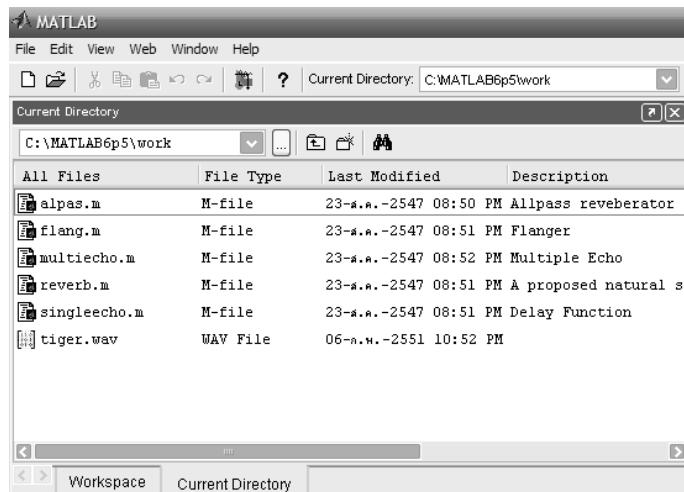


Fig. 3.1 การคลิกเลือกที่ Current Directory

- โดยการคลิกขวาที่ไฟล์.wav และเลือกที่ import data

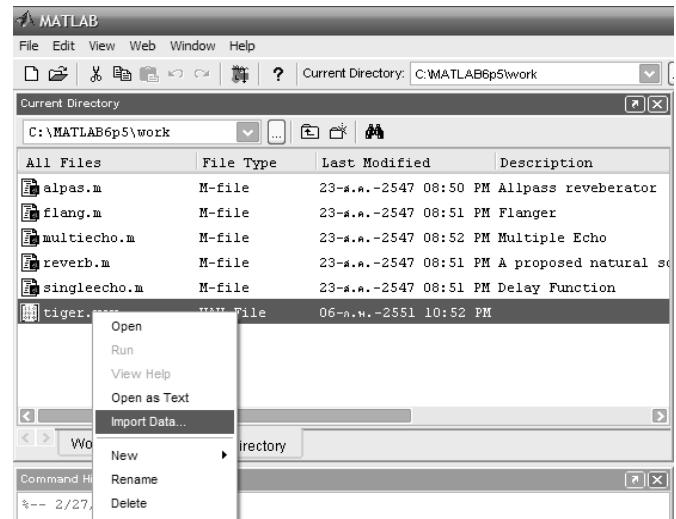


Fig. 3.2 การ import ไฟล์เพลง

- การเล่นเสียงต้นฉบับโดยการกดที่ปุ่ม Play Sound



Fig 3.2 การเล่นเสียงต้นฉบับ

- การดูข้อมูลเสียงในรูปของ Matrix

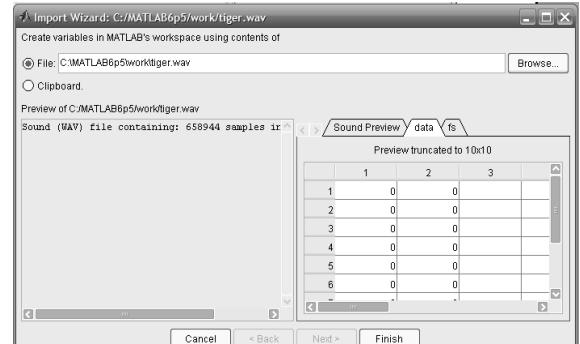


Fig. 3.3 ดูข้อมูลเสียงในรูปของ Matrix

- การดู sampling frequency ของเสียง

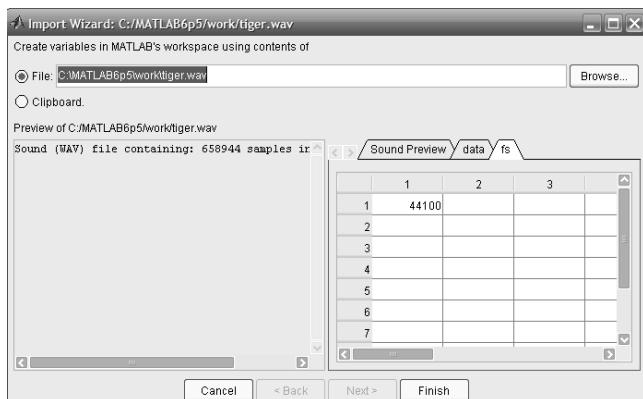


Fig. 3.4 ค่าของ fs ต้องมีค่าเท่ากับ 44,100

4. การใส่เอฟเฟกต์ให้กับไฟล์เสียง

- หน้าต่างของการพิมพ์ Code คำสั่ง

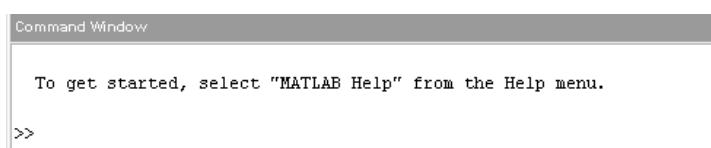


Fig. 4.1 หน้าต่างของการพิมพ์ Code คำสั่ง

- ประการตัวแปรที่ชื่อว่า “data” และคึ่งข้อมูลจาก structure นำเข้าใส่ใน data

```
Command Window
Using Toolbox Path Cac
To get started, select
>> data = tiger.data|
```

Fig. 4.2 การประการตัวแปร data

```
-0.1402 0.0045
-0.1444 0.0045
-0.1485 0.0046
-0.1525 0.0047
-0.1562 0.0047
-0.1594 0.0046
-0.1626 0.0046
-0.1654 0.0047
>>
```

Fig. 4.3 เมื่อประการตัวแปร data แล้ว

Workspace			
Name	Size	Bytes	Class
data	658944x2	10543104	double array
tiger	1x1	10543360	struct array

Fig. 4.4 เมื่อประการตัวแปร data แล้ว

- ประการตัวแปรที่ชื่อว่า “fa” และคึ่งข้อมูลจาก structure นำเข้าใส่ใน fa

```
>> fa= tiger.fs|
```

Fig. 4.5 การประการตัวแปร fa

```
>> fa= tiger.fs
fa =
44100
>>
```

Fig. 4.6 เมื่อประการตัวแปร fa แล้ว

Name	Size	Bytes	Class
data	658944x2	10543104	double array
fa	1x1	8	double array
tiger	1x1	10543360	struct array

Fig. 4.7 เมื่อประการตัวแปรแล้ว

5. การทดสอบเสียงโดยใช้ code

- โดยพิมพ์ code
- data คือ ตัวข้อมูลเสียง
- 8000 คือ ค่า delay
- 0.5 คือ ค่าแอมเพลจูดของไฟล์เสียง

```
>> y=singleecho(data,8000,0.5);
```

Fig. 5.1 การเขียน code

6. การเล่นเสียง

- โดยพิมพ์ว่า sound(y,fa);

```

>>
>> sound(y,fs);
>> |

```

Fig. 6.1 code เพื่อการเล่นเสียง

7. การพล็อตกราฟ

- โดยพิมพ์ว่า `plot(y);`

```

>> plot(y);
>>

```

Fig. 7.1 code เพื่อการ plot กราฟ

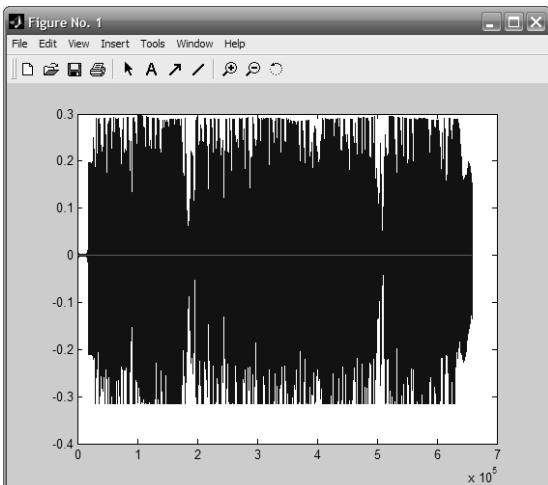


Fig. 7.2 ลักษณะกราฟที่ผสม effect เข้ากับไฟล์เสียง
ແລ້ວ

- ถ้าเป็น Effect Multiple Echo ก็ทำในแบบ
เดียวกัน เราสามารถผสมหลายๆ Effect เข้า
ด้วยกันได้ และสามารถปรับค่าต่างๆ ได้
ตามใจชอบ ต้องทดสอบฟังเสียงดู

ตัวอย่าง Code ไฟล์สร้างเสียง Effect กีตาร์

ไฟล์ singleecho.m

```

%Single Echo Function
% y = singleecho(x, R, a);
%
% Parameters:
% x is the input audio signal
% R is the delay in number of samples

```

% a specifies the attenuation in the echo

%

% Return value:

% y is the output signal

%

% Example:

% [x,fs,nbits] = wavread('dsp01.wav');

% y = singleecho(x,8000,0.5);

% wavplay(y,fs);

function y = singleecho(x, R, a);

xlen=length(x); %Calc. the number of samples in
the file

y=zeros(size(x));

% filter the signal

for i=1:1:R+1

 y(i) = x(i);

end

for i=R+1:1:xlen

 y(i)= x(i)+ a*x(i-R);

end;

ไฟล์ multipleecho.m

%Multiple Echo

% y = multiecho(x,R,a,N);

%

% Generates multiple echos R samples apart with
exponentially decaying amplitude

% Parameters:

% x is the input audio signal

% R is the delay in number of samples

% a specifies the attenuation in the echos

% N-1 is the total number of echos (If N = 0, an
infinite number of echos is produced)

%

% Return value:

% y is the output signal

%

% Example:

% [x,fs,nbits] = wavread('dsp01.wav');

% y = multiecho(x,8000,0.5,3);

% wavplay(y,fs);

function y = multiecho(x,R,a,N);

if (N == 0)

 num=[zeros(1,R),1];

 den=[1,zeros(1,R-1),-a];

else

 num=[1,zeros(1,N*R-1),-a^N];

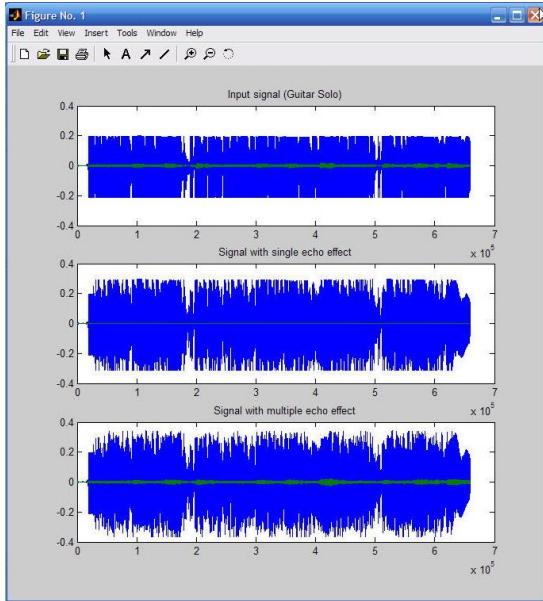
 den=[1,zeros(1,R-1),-a];

end

y=filter(num,den,x);

4. ผลการทดลอง

ผลการทดลองที่ได้สามารถแสดงได้อยู่ในรูปกราฟเสียงจากโปรแกรม MATLAB ได้ดังนี้



1. กราฟบนสุดคือ เสียงต้นฉบับ ในที่นี่มาจากการ Solo เพลงปลุกใจเลือป่า ของวง Big ass
2. เพลงปลุกใจเลือป่าที่ใส่เอฟเฟกต์ Single Echo แล้ว
3. เพลงปลุกใจเลือป่าต้นฉบับนำมาใส่เอฟเฟกต์ Multiple Echo โดยไม่ได้ใส่เอฟเฟกต์ Single Echo

5. สรุปผลการดำเนินงาน

ทางกลุ่มสามารถทำ Effect กีตาร์พื้นฐานได้ โดยศึกษาจากทฤษฎีของวิชา Signals and Systems โดยสามารถหา impulse response ตามที่ต้องการได้สามารถใช้งานได้จริง เพียงแต่ต้องใช้การ Code จากโปรแกรม MATLAB เท่านั้น Effect ที่ทำได้ยังเป็นแค่ Effect พื้นฐานเพียงแค่ 2 Effect เท่านั้น ซึ่งแท้จริงแล้วเสียงดนตรีสามารถใส่ Effect ได้หลายประเภท อีกทั้งเรายังสามารถคิดกัน Effect ใหม่ๆ ได้อ่องอีกด้วย ซึ่งสิ่งเหล่านี้ ต้องใช้ความรู้ในวิชา Digital Signal

Processing ซึ่งเป็นภาษาสตรีชั้นสูง nokหนีขอบเขตของโครงการนี้ ทางกลุ่มถ้ามีโอกาสจะได้ทำการศึกษาค้นคว้าต่อไป

6. เอกสารอ้างอิง

Sanjit K. Mitra . *Digital Signal Processing* . Third Edition . California , 2006 .

James H. McClellan, Ronald W. Schafer, Mark A. Yoder . *Signal Processing First* . First Edition . United States of America,2003 .

Alan V. Oppenheim, Alan S. Willsky, S. Hamid Nawab . *Signal and Systems* . Second Edition . United States of America , 1997 .

Simon Haykin , Barry Van Veen . *Signals and Systems* . Second Edition . United States of America , 2002 .

ลักษณ์ วุฒิศิทธิกลิจ . *MATLAB การประยุกต์ใช้งานทางวิศวกรรมไฟฟ้า* . พิมพ์ครั้งที่ 1 . กรุงเทพฯ : บริษัทค่านสุทธารการพิมพ์,พ.ศ. ๒๕๔๗ .

<http://www.musicdsp.org> , (วันที่ค้นข้อมูล : 18 มกราคม 2551).

<http://www.harmony-central.com> , (วันที่ค้นข้อมูล : 18 มกราคม 2551).