บทความวิจัย

โครงข่ายประสาทเทียมสำหรับการจำลองผลตอบสนองอิมพัลส์ของตู้ลำโพงกีตาร์แบบเวลาจริง

ธัญเทพ สิญจนาคม และ ศรวัฒน์ ชิวปรีชา*
ภาควิชาวิศวกรรมโทรคมนาคมคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 02 329 8324 อีเมล: sorawat.ch@kmitl.ac.th
DOI: 10.14416/j.kmutnb.2024.03.016
รับเมื่อ 1 กุมภาพันธ์ 2565 แก้ไขเมื่อ 7 เมษายน 2565 ตอบรับเมื่อ 5 พฤษภาคม 2565 เผยแพร่ออนไลน์ 29 มีนาคม 2567

บทคัดย่อ

งานวิจัยนี้นำเทคโนโลยีการประมวลผลสัญญาณดิจิทัลแบบเวลาจริง และโครงข่ายประสาทเทียมมาพัฒนาระบบที่สร้างผลตอบสนองอิมพัลส์ของลำโพงตู้กีตาร์ Marshall 1960A ตามลักษณะการติดตั้งไมโครโฟนตามที่ผู้ใช้กำหนด โดยโมเดลจะรับค่าเป็นประเภทไมโครโฟน ตำแหน่งของลำโพงที่ติดตั้งไมโครโฟน ระยะห่างระหว่างไมโครโฟนกับตู้ และมุมเอียง โมเดลโครงข่ายประสาทเทียมที่ผ่านการฝึกสอนสามารถสร้างผลตอบสนองอิมพัลส์สำหรับตู้ลำโพงได้ทั้งเสียงที่มีอยู่ในชุดข้อมูล และเสียงของการตั้งค่าที่ไม่มีอยู่ในชุดข้อมูลซึ่งเกิดจากการเรียนรู้ความสัมพันธ์ของข้อมูล เกณฑ์ที่ใช้ประเมินผลลัพธ์ที่ได้จากโมเดล คือ Cross-correlation, Error-to-signal Ratio, Power Spectral Density Error และ Magnitude-squared Coherence นอกจากนี้ มีการทดสอบการฟังคะแนนความคิดเห็นเฉลี่ยเพื่อพิจารณาความคล้ายคลึงของสัญญาณกีตาร์ที่ผ่านการจำลองเสียงลำโพง ผลการทดสอบชี้ว่าเสียงที่ผ่านการจำลองด้วยเอาต์พุตของโครงข่ายประสาทเทียมนั้นมีความใกล้เคียงกับเสียงที่จำลองด้วย IR จริงอย่างมาก เมื่อนำโมเดลนี้ไปสร้างเป็นดิจิทัลปลั๊กอินแล้วพบว่า มีประสิทธิภาพในการคำนวณที่รวดเร็วพอกับการทำงานแบบเวลาจริง การนำโมเดลนี้มาใช้งานนั้นไม่จำเป็นที่จะต้องเก็บข้อมูล IR ไว้ในคอมพิวเตอร์โดยตรงเหมือนกับการทำงานรูปแบบเดิม โมเดลนี้สามารถสร้าง IR ขึ้นมาทุกครั้งที่ผู้ใช้กำหนดค่าพารามิเตอร์ต่าง ๆ และการใช้ระบบดังกล่าวในงานผลิตเพลงจะทำให้ผู้ใช้สามารถปรับแต่งเสียงได้สะดวกเพราะจะได้ฟังเสียงความแตกต่างของการตั้งค่าต่าง ๆ ทันที โดยไม่ต้องโหลดไฟล์ IR ของการตั้งค่าแต่ละแบบไปมาเหมือนการทำงานแบบเดิม

คำสำคัญ: โครงข่ายประสาทเทียม ผลตอบสนองอิมพัลส์ การประมวลผลสัญญาณดิจิทัล ตู้ลำโพง การจำลองแบบเวลาจริง

การอ้างอิงบทความ: ธัญเทพ สิญจนาคม และ ศรวัฒน์ ชิวปรีชา, "โครงข่ายประสาทเทียมสำหรับการจำลองผลตอบสนองอิมพัลส์ของตู้ลำโพงกีตาร์แบบเวลาจริง ," *วารสารวิชาการพระจอมเกล้าพระนครเหนือ*, ปีที่ 34, ฉบับที่ 2, หน้า 1–13, เลขที่บทความ 242-245787, เม.ย.–มิ.ย. 2567.

วารสารวิชาการพระจอมเกล้าพระนครเหนือ ปีที่ 34 ฉบับที่ 2 เม.ย.–มิ.ย. 2567
The Journal of KMUTNB., Vol. 34, No. 2, Apr.–Jun. 2024

Research Article

Artificial Neural Networks for Real-Time Digital Emulation of Guitar Speaker Cabinet Impulse Response

Tantep Sinjanakhom and Sorawat Chivapreecha*
Department of Telecommunications Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

* Corresponding Author, Tel. 02 329 8324, E–mail: sorawat.ch@kmitl.ac.th DOI: 10.14416/j.kmutnb.2024.03.016
Received 1 February 2021 ; Revised 7 April 2022 ; Accepted 5 May 2022; Published online: 29 March 2024

Abstract

This research employs real- time digital signal processing technologies and a neural network to create a system capable of generating the impulse response (IR) of a Marshall 1960A guitar cabinet speaker depending on a user-specified microphone setup. The microphone type, location of the speaker on which the microphone is placed, the distance between the microphone and the cabinet, and off-axis angle are all used as inputs by the model. Since the network can learn the correlations between the microphone position inputs and the related IR outputs, the trained neural network model can create IR for the speaker cabinet for both sounds that exist in the dataset and sounds that do not exist in the dataset. Cross-correlation, error-to-signal ratio, power spectral density error, and magnitude- squared coherence were used to evaluate the output of the model. Mean Opinion Score (MOS) listening tests were performed to determine the similarity of the emulated guitar signals. According to the results, the emulated cabinet sounds were perceived to be nearly identical to the original sounds. The performance of the implemented real-time audio plugin is proved to be computationally efficient. Since raw IR data for each microphone setup does not need to be explicitly stored in the PC's memory, using it in music production work allows the user to change the settings while hearing the differences without having to redo the IR file loading procedure.

Keywords: Neural Networks, Impulse Response, Digital Signal Processing, Speaker Cabinet, Real-Time Emulation

Please cite this article as: T. Sinjanakhom and S. Chivapreecha, "Artificial neural networks for real-time digital emulation of guitar speaker cabinet impulse response," *The Journal of KMUTNB*, vol. 34, no. 2, pp. 1–13, ID. 242-245787, Apr.–Jun. 2024 (in Thai).

1. บทนำ

การหาโทนเสียงกีตาร์ที่เหมาะสมในงานมิกซ์เพลงเป็นเรื่องที่ท้าทายเทคนิคต่าง ๆ ถูกพัฒนาขึ้นเพื่อใช้ปรับคุณลักษณะของเสียงกีตาร์ไฟฟ้า หนึ่งในวิธีที่นิยม คือ การนำผลตอบสนองอิมพัลส์ (Impulse Response; IR) ของตู้ลำโพง (Speaker Cabinet) ที่มีผลตอบสนองความถี่ที่ต้องการมาทำการคอนโวลูชันกับเสียงกีตาร์ กระบวนการนี้จะถูกใช้ใน Digital Audio Workstation (DAW) ซึ่งเป็นซอฟต์แวร์ที่ใช้ในการทำเพลงซึ่งครอบคลุมการบันทึกเสียง ตัดต่อเสียง มิกซ์เสียง และมาสเตอร์ กระบวนการดังกล่าวต้องใส่เอฟเฟกต์เสียงช่วยปรับแต่ง โดยเอฟเฟกต์เสียงอยู่ในรูปแบบซอฟต์แวร์เสริมที่เรียกว่าดิจิทัลปลั๊กอิน

กระบวนการปรับแต่งเสียงนี้เป็นการทดลองซ้ำ ๆ เพื่อหาเสียงกีตาร์ที่เหมาะกับเพลง ผู้ใช้ต้องโหลดไฟล์บันทึกเสียง IR ของแต่ละการตั้งค่าเข้า IR Loader ทีละไฟล์เพื่อฟังเสียงการจำลองใน DAW ซึ่งใช้เวลานานกว่าจะได้เสียงที่พอใจ โดยการตั้งค่าดังกล่าวนั้นประกอบไปด้วยรุ่นของลำโพง ไมโครโฟนที่ใช้อัด รวมไปถึงตำแหน่งที่ตั้งไมโครโฟน โมเดลที่นำเสนอเป็นเครื่องมือที่ครบครันในตัวเดียว (All-in-one) โดยสามารถสร้าง IR ได้ทันทีเมื่อผู้ใช้กำหนดพารามิเตอร์ประเภทไมโครโฟน ตำแหน่งของลำโพงที่ติดตั้ง ระยะห่างระหว่างไมโครโฟนกับตู้ และมุมเอียง ซึ่งการทำงานรูปแบบนี้จะช่วยให้ได้ยินความแตกต่างของโทนเสียงขณะปรับเสียงได้ชัดเจนมากกว่าวิธีการเดิม เพราะการโหลด IR ทีละไฟล์แบบเดิมอาจเกิดปัญหา เช่น การโหลด IR ใหม่เสร็จผู้ใช้อาจลืมเสียงของ IR ก่อนหน้าไปแล้ว นอกจากนี้ผู้ใช้ไม่จำเป็นต้องมีพื้นที่ในหน่วยความจำคอมพิวเตอร์สำหรับจัดเก็บไฟล์ IR ดิบจำนวนมาก

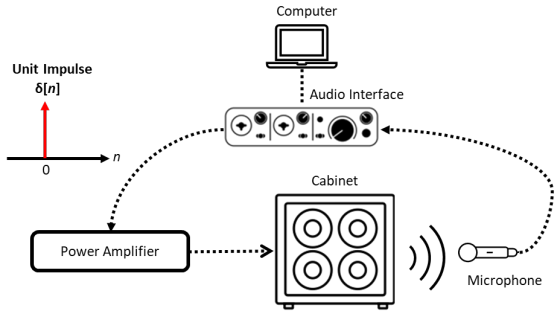
ในช่วงสองถึงสามปีที่ผ่านมามีการศึกษาเกี่ยวกับการสร้าง IR ด้วยโครงข่ายประสาทเทียมบ้างแล้ว ส่วนใหญ่มุ่งเน้นการสร้าง Room Impulse Response (RIR) เช่น IR-GAN [1] เป็นหนึ่งในงานวิจัยที่ใช้การเรียนรู้เชิงลึกเพื่อสังเคราะห์ RIR เพื่อปรับปรุงระบบรู้จำเสียงพูด เนื่องจาก GAN ต้องใช้ GPU ช่วยคำนวณจึงจะทำงานได้ จึงไม่เหมาะกับงานนี้เนื่องจากงานประมวลผลเสียงส่วนใหญ่ดำเนินการภายใน DAW ที่ขับเคลื่อนด้วย CPU เป็นหลัก นอกจากนี้มีงานวิจัยที่พัฒนาโครงข่ายประสาทเทียมคอนโวลูชัน (Convolutional Neural Network; CNN) เพื่อใช้สร้างผลตอบสนองอิมพัลส์ทางอะคูสติก (Acoustic Impulse Response; AIR) ที่ได้ความแม่นยำสูง [2] แต่ก็มีความซับซ้อนในการคำนวณสูงเช่นกันซึ่งไม่เหมาะกับการนำมาใช้งานในระบบเวลาจริง ส่วนงานวิจัยที่มีความใกล้เคียงกับงานที่นำเสนอนี้คือปลั๊กอินเสียง Neural Reverberator [3] ซึ่งใช้โมเดล Autoencoder สังเคราะห์ RIR แต่ IR ทั้งหมดถูกสร้างไว้ล่วงหน้าและเก็บเป็นตาราง Look-up ซึ่งต้องมีพื้นที่หน่วยความจำสูง ส่วนวิธีอื่น ๆ ในการจำลองเอฟเฟกต์เสียงและแอมพลิฟายเออร์ได้ถูกนำเสนอ เช่น [4]–[6] ซึ่งส่วนใหญ่จะจำลองเสียงโดยตรง กล่าวคือการใช้สัญญาณที่ไม่ผ่านการปรับแต่งใด ๆ เป็นอินพุตและสัญญาณเอาต์พุตของแอมพลิฟายเออร์เป็นค่าเป้าหมาย โดยใช้โมเดล CNN แบบ WaveNet และโครงข่ายประสาทเทียมแบบเกิดซ้ำ (Recurrent Neural Network; RNN) ซึ่งมีความแม่นยำสูง แต่เมื่อใช้งานแบบเวลาจริง (Real-time) จะมีปัญหา CPU Overload

แนวคิดของการศึกษานี้แตกต่างจากงานวิจัยอื่น แต่ได้ผลลัพธ์ในลักษณะเดียวกัน เพราะเป็นการผสมผสานของการเรียนรู้ด้วยเครื่อง (Machine Learning) สำหรับการสร้าง IR และการประมวลสัญญาณดิจิทัลแบบเดิม ภาพรวมของระบบคือ โครงข่ายประสาทเทียมรับค่าประเภทของไมโครโฟนและการจัดวางไมโครโฟน และสร้าง IR แบบเวลาจริง โมเดลได้รับการฝึกสอน (Train) ด้วย IR ของตู้กีตาร์โดยใช้ชุดข้อมูลขนาดเล็ก อย่างไรก็ตาม โครงข่ายสามารถเรียนรู้และสร้างเสียงของการตั้งค่าไมโครโฟนที่ไม่มีอยู่ในชุดข้อมูลได้ นอกจากนี้ โมเดลนี้ยังสามารถฝึกเพิ่มเติมสำหรับ IR ประเภทอื่น ๆ ได้อีก

2. วัสดุ อุปกรณ์และวิธีการวิจัย

2.1 การวัดผลตอบสนองอิมพัลส์

IR คือ ผลตอบสนองที่กำหนดคุณลักษณะของระบบเชิงเส้นไม่แปรผันตามเวลา (Linear Time-Invariant; LTI) เมื่อนำ IR ของระบบมาทำ คอนโวลูชันกับสัญญาณเสียงจะสามารถเปลี่ยนแปลงคุณลักษณะเสียงได้ เช่น การใช้สัญญาณเสียง



รูปที่ 1 ไดอะแกรมการบันทึก IR ของตู้ลำโพงกีตาร์

IR ของโบสถ์เพื่อจำลองความก้องสะท้อน ในทำนองเดียวกันการนำเสียงบันทึกกีตาร์ผ่านกล่อง Direct Injection (DI) ซึ่งจะเป็นเสียงที่ไม่ผ่านการปรุงแต่งใด ๆ มาทำคอนโวลูชันกับ IR ของตู้ลำโพงจะสร้างเสียงราวกับต่อกีตาร์เข้าลำโพงนั้น ๆ กระบวนการนี้เรียก Cabinet Simulation ซึ่งทำให้การทดลองกับลำโพงรุ่นต่าง ๆ มีความสะดวก โดยไม่จำเป็นต้องนำลำโพงจริงมาบันทึก การวัด IR ของระบบ LTI สามารถวัดได้หลายวิธี [7] วิธีที่ง่ายที่สุด คือ การส่งสัญญาณยูนิตอิมพัลส์ผ่านระบบและบันทึกดังแสดงในรูปที่ 1

2.2 การทำคอนโวลูชันแบบเร็ว

คอนโวลูชันเป็นวิธีการประมวลผลสัญญาณที่สามารถนำมาประยุกต์ใช้เพื่อจำลองเสียงกีตาร์ที่บันทึกผ่านกล่อง DI ให้มีเสียงเหมือนต่อเข้ากับตู้ลำโพง โดยทั่วไปคอนโวลูชันสามารถคำนวณได้ในโดเมนเวลา เมื่อใช้สัญญาณอินพุต $x[n]$ ความยาว N จุด และสัญญาณ IR $h[n]$ ความยาว M จุด สัญญาณเอาต์พุต $y[n]$ ที่ได้จะยาว $\mathrm{N}+\mathrm{M}-1$ [8] ทั้งนี้การคำนวณโดเมนเวลาไม่สามารถทำได้ในแบบเวลาจริง เนื่องจากความซับซ้อนในการคำนวณเพิ่มขึ้นตามความยาวของสัญญาณ IR ส่งผลให้มีการคูณและบวกจำนวนมาก วิธีการที่ช่วยแก้ปัญหานี้ได้ คือดำเนินการในโดเมนความถี่แทน หลักการคือใช้ Fast Fourier Transform (FFT) แปลงสัญญาณอินพุตและสัญญาณ IR จากนั้นนำผลการแปลงทั้งสองมาคูณกันและแปลงกลับด้วย Inverse Fast Fourier Transform (IFFT) เป็นสัญญาณในโดเมนเวลา หลักการนี้เรียกว่าคอนโวลูชันแบบเร็ว (Fast Convolution) เป็นวิธีที่ช่วยลดความซับซ้อนและเร่งความเร็วของกระบวนการได้ถึงร้อยเท่า [8] อย่างไรก็ตาม ปัญหาความยาวของสัญญาณอาจเกิดขึ้นได้ในกระบวนการนี้ กล่าวคือผลลัพธ์จากการทำคอนโวลูชันปกติจะมีความยาว $\mathrm{N}+\mathrm{M}-1$ ดังนั้น หากใช้การแปลง N-point IFFT สัญญาณเอาต์พุตจะไม่สมบูรณ์ แนวทางการแก้คือใช้คอนโวลูชันแบบวงกลมซึ่งจะนำข้อมูลที่เกินขนาด IFFT วนกลับมาบวกด้วยหลักการ Overlap-add

2.3 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียมคือรูปแบบหนึ่งของระบบปัญญาประดิษฐ์ ซึ่งเป็นอัลกอริทึมที่พัฒนาขึ้นเพื่อเลียนแบบกลไกระบบประสาทของสมองมนุษย์ ประกอบไปด้วยชั้นอินพุต ชั้นซ่อน และชั้นเอาต์พุต โครงข่ายนั้นสามารถนำไปใช้ได้กับงานการเรียนรู้ที่หลากหลาย เช่น การจำแนก การถดถอย และอื่น ๆ โครงสร้างที่ใช้ในงานวิจัยนี้ คือโครงข่ายแบบเพอร์เซ็ปตรอนหลายชั้น (Multilayer Perceptron; MLP) ประกอบด้วยโหนดจำนวนมากเชื่อมโยงกันในแต่ละชั้น ประมวลผลด้วยการนำอินพุตจากหน่วยก่อนหน้ามาดำเนินการคำนวณภายใน ผลรวมถ่วงน้ำหนักของแต่ละชั้นที่สามารถเขียนเป็นสมการได้ดังนี้

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l \tag{1}$$

$$a_j^l = f(z_j^l) \tag{2}$$

w_{jk}^l คือ ค่าน้ำหนักสำหรับการเชื่อมต่อจากโหนดที่ k ในชั้นที่ $(l-1)$ ถึงโหนดที่ j ในชั้นที่ l ส่วน b_j^l คือ ค่าไบแอสของโหนดที่ j ในชั้นที่ l และ a_j^l คือ ผลจากการใช้ฟังก์ชันกระตุ้น $f(\ )$ กับโหนดที่ j ในชั้นที่ l สำหรับชั้นอินพุตนั้นใช้ x_k แทน a_k^{l-1} ในสมการที่ (1) และสำหรับชั้นเอาต์พุตนั้นใช้ y_k แทน a_j^l ในสมการที่ (2) วิธีการที่ใช้ในการสอนโครงข่าย คือ หลักการแพร่ย้อนกลับ (Backpropagation) ซึ่งเป็นการคำนวณความชันของฟังก์ชันการสูญเสียเทียบเมื่อกับค่าน้ำหนักของแต่ละโหนด เมื่อ L คือ จำนวนชั้นและ ε คือ ฟังก์ชันการสูญเสีย สมการที่ (3) ใช้คำนวณค่าความผิดพลาดที่ชั้นเอาต์พุต และ

สมการที่ (4) ใช้สำหรับการคำนวณค่าความผิดพลาดในชั้นต่อ ๆ ไป

$$\delta_j^L = \frac{\partial \varepsilon}{\partial a_j^L} f'(z_j^L) \quad (3)$$

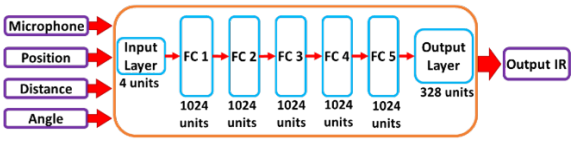
$$\delta^l = \left((w^{l+1})^T \partial^{l+1}\right) \odot f'(z^l) \quad (4)$$

เมื่อ $1 = L - 1, L - 2,...,2$ จากนั้นจะสามารถหาค่าความชันของฟังก์ชันการสูญเสียเมื่อเทียบกับค่าน้ำหนัก $\frac{\partial \varepsilon}{\partial w_{jk}^L} = a_k^{l-1}\delta_j^l$ และค่าความชันเมื่อเทียบกับค่าไบแอส $\frac{\partial \varepsilon}{\partial b_j^L} = \delta_j^l$ ของโหนดต่าง ๆ ในโครงข่าย และนำค่าเหล่านี้เป็นปรับน้ำหนักและไบแอสเพื่อลดความผิดพลาดให้น้อยลงในแต่ละรอบของการฝึก นอกจากนี้โครงข่ายประสาทเทียมประเภทอื่น ๆ ที่มักใช้สำหรับงานที่เกี่ยวข้องกับเสียง ได้แก่ CNN และ RNN อัลกอริทึมของ CNN สามารถสกัดคุณลักษณะของอินพุตได้ดีโดยการรวมตัวกรองที่เรียนรู้ได้ ในขณะที่ RNN นั้นเหมาะสมกับข้อมูลแบบลำดับหรืออนุกรมเวลา ทั้งนี้โครงข่ายทั้งสองประเภทมีความซับซ้อนในการคำนวณที่สูงกว่า MLP ส่งผลให้เวลาใช้งานจริงใน DAW ซึ่งใช้เฉพาะ CPU คำนวณเท่านั้น ไม่มี GPU มาช่วยคำนวณ จะส่งผลให้ใช้ CPU Resource จำนวนมากจนเกิดการ Overload ได้

2.4 ระบบที่นำเสนอ

โมเดลที่เสนอประกอบด้วย 2 ส่วน ส่วนแรก คือ การสังเคราะห์ IR ตู้ของลำโพงด้วยโครงข่ายประสาทเทียม ส่วนที่สอง เป็นการนำค่าน้ำหนักและไบแอสของโครงข่ายดังกล่าวมาใช้กับดิจิทัลปลั๊กอินซึ่งจะนำ IR ที่ทำนายมาทำ คอนโวลูชันแบบเร็วกับสัญญาณเสียงใน DAW

บทความนี้เสนอโครงข่ายประสาทเทียมที่เหมาะสมกับการทำงานแบบเวลาจริง ดังนั้นสถาปัตยกรรมโครงข่ายจึงเป็น MLP ที่มีชั้นซ่อนเพียง 5 ชั้น แต่ละชั้นมี 1,024 โหนด ฟังก์ชันกระตุ้นที่ใช้คือ ReLU (Rectified Linear Unit) โดยชั้นอินพุตรับ 4 ค่าคุณสมบัติ ได้แก่ ประเภทของไมโครโฟน ตำแหน่งการติดตั้งตามลำโพง ระยะห่าง และมุมเอียง



รูปที่ 2 สถาปัตยกรรมของ MLP ที่ออกแบบ

ชั้นเอาต์พุตจะทำนาย IR ความยาว 20.5 มิลลิวินาที หรือ 328 จุดที่สอดคล้องกับอินพุต อัตราการสุ่มตัวอย่าง (Sampling Frequency: Fs) ของ IR ที่โครงข่ายทำนายคือ 16 กิโลเฮิรตซ์ ซึ่งจะต้องเพิ่มอัตราสุ่ม (Upsampling) ภายหลังเพื่อให้ตรงกับ Fs ของ DAW

รูปที่ 2 แสดงสถาปัตยกรรมของ MLP ที่นำเสนอ ลักษณะการรับอินพุตจะคล้ายกับโมเดลที่ถูกนำเสนอใน [9] ที่เป็นการสร้าง RIR จากค่ามิติของห้อง ตำแหน่งผู้ฟังและลำโพง และเวลาเสียงก้อง ทั้งนี้โครงสร้างส่วนอื่นจะแตกต่างกันตรงกันข้ามกับโครงข่ายประสาทเทียมแบบ CNN หรือ RNN โครงข่ายประสาทแบบ MLP สามารถคำนวณการถดถอยที่ซับซ้อนปานกลางได้โดยไม่ใช้ทรัพยากรการคำนวณสูง

2.5 ฟังก์ชันการสูญเสียและเกณฑ์การวัด

โมเดล MLP ได้รับการฝึกสอนเพื่อลดข้อผิดพลาดระหว่างสัญญาณผลตอบสนองอิมพัลส์ที่คาดการณ์ y_p และผลตอบสนองอิมพัลส์เป้าหมาย y_t และ เพื่อหาฟังก์ชันการสูญเสียที่เหมาะสมต่อการใช้ฝึกสอนที่สุด ได้มีการเปรียบเทียบฟังก์ชันการสูญเสียสองฟังก์ชันได้แก่ ค่าครึ่งหนึ่งของค่าผิดพลาดกำลังสองเฉลี่ย (Half of Mean Square Error; HMSE) ในสมการที่ (5) ซึ่งเป็นฟังก์ชันเริ่มต้นใน Deep Learning Toolbox ของ MATLAB และฟังก์ชันผลรวมระหว่างอัตราส่วนค่าผิดพลาดต่อสัญญาณ (Error-to-Signal Ratio; ESR) ในสมการที่ (6) กับฟังก์ชันสูญเสียงไฟตรง (DC Loss) ในสมการที่ (7) [10]

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l \quad (5)$$

$$\varepsilon_{ESR} = \frac{\sum_{n=0}^{N-1}(\left|y_t[n]-y_p[n]\right|^2)}{\sum_{n=0}^{N-1}(\left|y_t[n]\right|^2)} \quad (6)$$

$$\varepsilon_{DC} = \frac{\left|\frac{1}{N}\sum_{n=0}^{N-1}(y_t[n]-y_p[n])\right|^2}{\frac{1}{N}\sum_{n=0}^{N-1}\left|y_t[n]\right|^2} \quad (7)$$

$$\varepsilon = \varepsilon_{ESR} + \varepsilon_{DC} \quad (8)$$

ESR คือ ค่าสัญญาณผิดพลาดกำลังสองเมื่อเทียบกับพลังงานของสัญญาณทั้งหมดและ DC Loss ระบุค่าความแตกต่างของ DC Offset ระหว่างสัญญาณเป้าหมายและสัญญาณเอาต์พุตของโครงข่าย ค่ารากที่สองของค่าผิดพลาดกำลังสองเฉลี่ย (Root Mean Squared Error; RMSE) ถูกใช้เป็นเกณฑ์การเปรียบเทียบผลลัพธ์ของโมเดลที่ฝึกด้วยฟังก์ชันการสูญเสียทั้งสอง โมเดลที่ฝึกด้วยผลรวมของ ESR และ DC Loss ในสมการที่ (8) ให้อัตราความผิดพลาดที่ 0.0194 ซึ่งต่ำกว่าค่าของโมเดลที่ฝึกโดยใช้สมการที่ (5) ซึ่งเท่ากับ 0.0219

เพื่อวัดความแม่นยำของโมเดลแบบจำลองลำโพงเพิ่มเติม มีการใช้เกณฑ์การวัดความเหมือนของ IR ทั้งในโดเมนเวลาและความถี่ ประกอบด้วยฟังก์ชันสหสัมพันธ์ไขว้ (Cross-Correlation), ESR ความคลาดเคลื่อนของความหนาแน่นสเปกตรัมกำลัง (Power Spectral Density Error; PSDE) และความสอดคล้องของขนาดกำลังสอง (Magnitude-Squared Coherence; MSC) โดยที่ค่า Cross-correlation จะระบุว่าสัญญาณมีความเหมือนกันมากเพียงใด ฟังก์ชันนี้คำนวณผลรวมของผลคูณของแต่ละจุดตัวอย่างเมื่อสัญญาณหนึ่งเลื่อนไปด้วย lag ที่ k ดังที่แสดงในสมการที่ (9) สัญญาณจะเหมือนกันอย่างเห็นได้ชัดถ้าค่าที่ผ่านการ Normalize อยู่ใกล้หรือเท่ากับ 1

$$Corr_{pt}[n] = \sum_{k=-N}^{N} y_p[n+k]y_t[k] \quad (9)$$

ในกรณีของการประเมินโดเมนความถี่ ฟังก์ชันแรกคือ PSDE ซึ่งระบุความแตกต่างระหว่างความหนาแน่นของสเปกตรัมกำลังของสัญญาณที่ทำนายและสัญญาณเป้าหมายตามที่แสดงในสมการที่ (10)

$$PSDE(f) = \left|P_t(f) - P_p(f)\right| \quad (10)$$

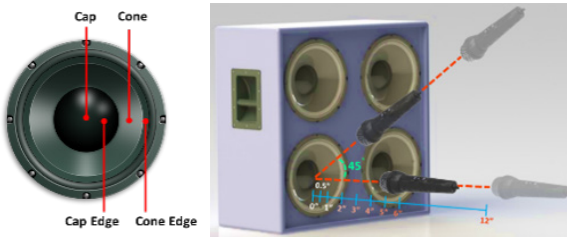
P_p และ P_t คือ ความหนาแน่นสเปกตรัมกำลังของ IR ที่โมเดลทำนายและของเป้าหมายตามลำดับ ค่าเฉลี่ยของความแตกต่างสัมบูรณ์ที่ได้รับตามแกนความถี่จะใช้เพื่อหาข้อผิดพลาดโดยรวม เพื่อประเมินความสัมพันธ์ระหว่างองค์ประกอบความถี่เพิ่มเติม ฟังก์ชัน MSC ถูกใช้คำนวณความสอดคล้องเชิงความถี่ มีค่าในช่วง 0 ถึง 1 สามารถเขียนเป็นสมการได้ดังนี้

$$MSC_{pt}(f) = \frac{\left|P_{pt}(f)\right|^2}{\left|P_{pp}(f)P_{tt}(f)\right|} \quad (11)$$

โดยที่ P_{pp} และ P_{tt} เป็นค่าสหสัมพันธ์อัตโนมัติของความหนาแน่นสเปกตรัมกำลังของ IR ที่โมเดลทำนายและของเป้าหมายตามลำดับ ส่วน P_{pt} คือ ค่าความหนาแน่นสเปกตรัมกำลังของค่าสหสัมพันธ์ระหว่างสัญญาณที่คาดการณ์และสัญญาณเป้าหมาย หากค่า MSC ที่แต่ละความถี่มีค่าใกล้เคียงหรือเท่ากับ 1 แสดงว่าสัญญาณ IR ที่คาดการณ์นั้นสอดคล้องกับ IR เป้าหมายในโดเมนความถี่เป็นอย่างดี

2.6 ฐานข้อมูล

ชุดข้อมูลที่ใช้สำหรับการฝึกสอนและทดสอบโมเดล คือ Redwirez Free IR Pack [11] เป็นชุดข้อมูลแบบโอเพนซอร์ส ภายในชุดข้อมูลจะมี IR ของตู้ลำโพงกีตาร์ Marshall รุ่น 1960A ที่ใช้ดอกลำโพงยี่ห้อ Celestion รุ่น G12M ขนาด 12 นิ้ว กำลังขับ 25 วัตต์ จำนวน 4 ดอก ไฟล์บันทึกเสียงถูกกำหนดไว้ด้วยประเภทของไมโครโฟน ตำแหน่งของลำโพงที่ติดตั้งไมโครโฟน ระยะห่างระหว่างไมโครโฟนกับตู้ และมุมเอียงตามลำดับแบบแยกอิสระจากกัน ดังนั้นงานวิจัยจึง

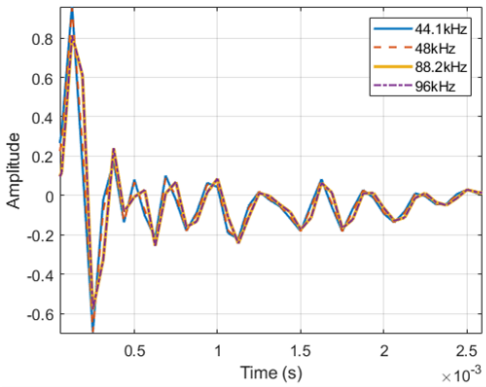


(ก) (ข)

รูปที่ 3 การวางไมโครโฟนและลำโพง (ก) ชื่อตำแหน่งของดอกลำโพง (ข) ตำแหน่งที่ติดตั้งไมโครโฟน

ได้นำข้อกำหนดดังกล่าวทั้งประเภทของไมโครโฟน ตำแหน่งของลำโพงที่ตั้งไมโครโฟน ระยะห่าง และมุมเอียงมาเป็นข้อมูลอินพุตให้กับโครงข่ายและใช้ข้อมูล IR ที่สอดคล้องกับค่าอินพุตดังกล่าวมาเป็นค่าเป้าหมายที่ให้โครงข่ายเรียนรู้ ไมโครโฟนมี 4 รุ่น ได้แก่ Shure SM57, Royer R121, Sennheiser MD421 และ Neumann KM84 มีระยะห่างระหว่างไมโครโฟนกับดอกลำโพงมี 9 ระยะได้แก่ 0, 0.5, 1, 2, 3, 4, 5, 6, และ 12 นิ้ว รูปที่ 3 (ก) แสดงตำแหน่งต่าง ๆ ของดอกลำโพงที่ได้มีการตั้งไมโครโฟนเพื่อบันทึกเสียงซึ่งมี 5 ตำแหน่ง ได้แก่ ฝาครอบ (Cap) ขอบฝาครอบ (Cap Edge) กรวย (Cone) ขอบกรวย (Cone Edge) และระหว่างดอกลำโพง (Center) ในทุกตำแหน่งของลำโพง ไมโครโฟนจะถูกตั้งไว้ที่ 0 องศา (On-axis) ยกเว้นตำแหน่ง Cap และ Cap Edge จะมีการเพิ่มตำแหน่งการเอียงนอกแกน (Off-axis) ที่ 45 องศา ซึ่งให้ลักษณะเสียงที่ต่างออกไป ถึงแม้จะมีข้อมูลของการตั้งไมโครโฟนแบบ Off-axis เพียง 2 ตำแหน่ง โครงข่ายสามารถเรียนรู้ความสัมพันธ์ของเสียงที่บันทึกที่ On-axis และ Off-axis และประยุกต์ใช้ทำนาย IR แบบ Off-axis ของตำแหน่งอื่นที่ไม่มีในชุดข้อมูลได้ รูปที่ 3 (ข) แสดงภาพรวมการตั้งไมโครโฟนในรูปแบบต่าง ๆ

IR ในชุดข้อมูลถูกบันทึกไว้ที่ความถี่ Fs ที่ 4 ความถี่ ได้แก่ 44.1 กิโลเฮิรตซ์ 48 กิโลเฮิรตซ์ 88.2 กิโลเฮิรตซ์ และ 96 กิโลเฮิรตซ์ ไฟล์ทั้งหมดถูกลดอัตราลง (Downsampling) ให้เหลือ 16 กิโลเฮิรตซ์ เท่ากัน ทั้งนี้เสียงบันทึก IR ของการตั้งไมโครโฟนที่เหมือนกันทุกประการแต่ถูกบันทึกที่คนละ



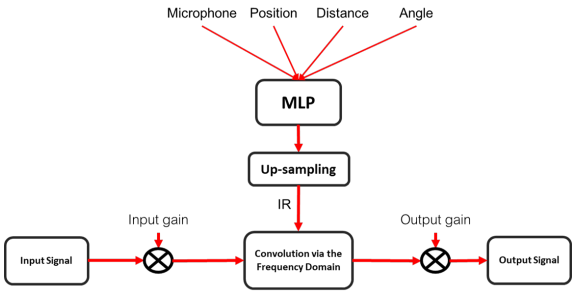
รูปที่ 4 การเปรียบเทียบ IR ที่ลดอัตราการสุ่มจาก Fs ต่าง ๆ

ความถี่ Fs เมื่อถูกลดอัตราสุ่มตัวอย่างแล้วจะมีลักษณะที่แตกต่างกันเล็กน้อย ดังนั้น การบันทึกเสียงจาก Fs ทั้ง 4 จึงถูกรวมอยู่ในชุดการฝึก และถือได้ว่าเป็นการเพิ่มข้อมูล (Data Augmentation) ซึ่งจะช่วยลดโอกาสการเกิด Overfitting ชุดข้อมูลนี้มีไฟล์เสียง IR ทั้งหมด 1,592 ไฟล์ รูปที่ 4 แสดงตัวอย่างความแตกต่างระหว่าง IR ที่บันทึกจาก Fs ที่แตกต่างกันหลังจากลดอัตราการสุ่มตัวอย่าง

ความยาวของ IR ในชุดข้อมูลเดิมทีจะมีความยาว 0.5 วินาที แต่ผู้วิจัยได้ตัดทอนให้เหลือ 20.5 มิลลิวินาที ซึ่งเป็นความยาวที่คล้ายกับของผลิตภัณฑ์เชิงพาณิชย์เช่น Suhr PT15 IR [12] เนื่องจาก IR ที่ยาวกว่านี้ไม่มีความแตกต่างในคุณภาพเสียงอย่างมีนัยสำคัญ ประโยชน์ที่ได้จากการลดความยาว IR คือการคำนวณที่เร็วขึ้น

2.7 การฝึกสอนโมเดล

กระบวนการฝึกสอนทั้งหมดดำเนินการด้วยอุปกรณ์คอมพิวเตอร์ที่มี CPU รุ่น 8th Generation Intel Core i7 หน่วยความจำ 16 GB และ GPU รุ่น NVIDIA GeForce MX150 ชุดข้อมูลแบ่งออกเป็น 80% (1,274 ไฟล์) ของชุดการฝึกและ 20% (318 ไฟล์) สำหรับการทดสอบ โมเดลถูกฝึกทั้งหมด 5,000 รอบโดยใช้ Adam Optimizer [13] อัตราการเรียนรู้เริ่มที่ 0.001 และลดลง 0.5 เท่า ทุก ๆ 200 รอบ ฟังก์ชันการสูญเสียในสมการที่ (8) ถูกใช้เพื่อลดข้อผิดพลาด และใช้เวลาประมาณ 20 นาทีในการฝึกสอน



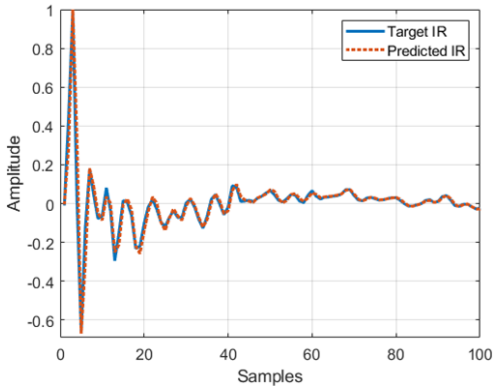
รูปที่ 5 ไดอะแกรมการเดินสัญญาณในดิจิทัลปลั๊กอิน

2.8 การประยุกต์ใช้งานแบบเวลาจริง

จุดประสงค์ของโมเดลที่ผ่านการฝึกสอนเรียบร้อยแล้ว คือ เพื่อให้สามารถใช้งานแบบเวลาจริงสำหรับการบันทึกเสียง มิกซ์เสียง มาสเตอร์ริ่ง และแอปพลิเคชันทางดนตรีอื่น ๆ เพื่อให้บรรลุสิ่งนี้ กระบวนการสังเคราะห์ IR และคอนโวลูชันจะต้องมีประสิทธิภาพในการคำนวณ รูปที่ 5 แสดงไดอะแกรมโดยรวมซึ่งนำไปใช้กับปลั๊กอินเสียงที่สร้างด้วย Audio Toolbox ของซอฟต์แวร์ MATLAB โมเดลจะรับ 4 อินพุต ที่จะกำหนดคุณลักษณะของ IR หลังจากที่โครงข่าย MLP คำนวณและสังเคราะห์ IR เสร็จจะนำไป Upsampling เพื่อให้ตรงกับความถี่ Fs ของ DAW หลังจากนั้น ระบบจะทำ Fast Convolution ระหว่าง IR และสัญญาณเสียงอินพุตโดยมี Partition Length อยู่ที่ 1024 จุดตัวอย่างและอัตราส่วนการ Overlap คือ 50% สุดท้าย สัญญาณจะถูกคูณด้วยอัตราขยายก่อนที่จะเล่นไปยังเอาต์พุตของ DAW

3. ผลการทดลอง

เบื้องต้นได้เปรียบเทียบการฝึก MLP ที่มีจำนวนชั้นซ่อนต่าง ๆ ตารางที่ 1 แสดงผลค่าผิดพลาดที่คำนวณด้วยสมการที่ (8) ในการฝึกโมเดลที่มีชั้นซ่อนตั้งแต่ 2 ถึง 7 ชั้นและระยะเวลาการคำนวณ (Runtime) ของ Forward Pass โมเดลที่มี 6 ชั้นซ่อนขึ้นไปนั้นถึงแม้จะมีความแม่นยำที่มากแต่มีปัญหาตอนสร้างเป็นปลั๊กอินเนื่องจากพื้นที่ Heap Memory ของ Compiler ไม่พอ จึงเลือกโมเดล 5 ชั้นซ่อนเพราะความแม่นยำดีที่สุดในขณะที่สร้างเป็นปลั๊กอินได้และใช้โหลด CPU เพียง 6% ซึ่งถือว่าใช้งานได้



รูปที่ 6 การเปรียบเทียบ IR ของไมโครโฟน Royer R121 ที่ตั้ง 3 นิ้ว จาก Cap Edge มีการเอียง 45 องศา

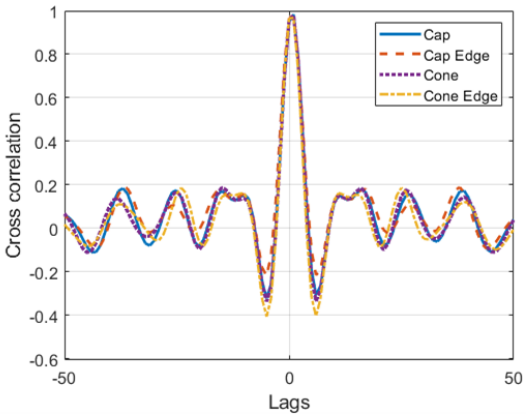
ตารางที่ 1 ผลการทดลองฝึก MLP ที่จำนวนชั้นซ่อนต่าง ๆ

Hidden Layers	Training Loss	Runtime (ms)
2	0.0627	2.932
3	0.0442	3.429
4	0.0386	3.822
5	0.0319	4.226
6	0.0239	4.628
7	0.0176	5.197

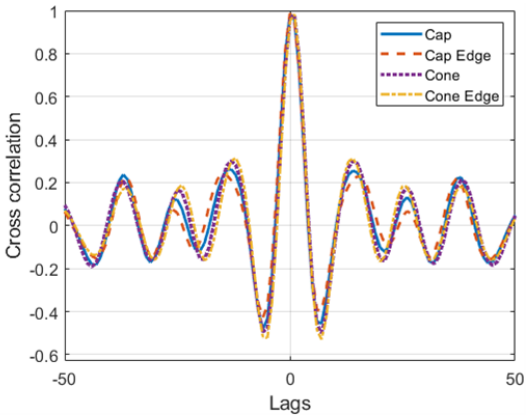
ค่าความผิดพลาดของการฝึกและการทดสอบของโมเดลที่มี 5 ชั้นซ่อนเท่ากับ 0.0319 และ 0.0413 ตามลำดับ ค่าความผิดพลาดเหล่านี้ใช้ประเมินความคลาดเคลื่อนของ IR ที่มี Fs เท่ากับ 16 กิโลเฮิรตซ์ อย่างไรก็ตาม เกณฑ์วัดอื่น ๆ จะใช้วัด IR ที่มีค่า Fs ที่สูงกว่าและเป็นมาตรฐานในงานด้านเสียงดนตรี และสามารถใช้งานได้ใน DAW ได้ รูปที่ 6 แสดงการเปรียบเทียบระหว่าง IR เป้าหมายกับ IR ที่คาดการณ์เป็นตัวอย่างจากการตั้งค่าไมโครโฟนเป็น Royer R121 จากรูปสามารถสังเกตได้ว่า IR ที่โครงข่ายคาดการณ์นั้นมีความใกล้เคียงกับ IR เป้าหมาย

3.1 การประเมินโมเดล

โมเดลที่ฝึกสอนเสร็จแล้วถูกประเมินประสิทธิภาพของแบบจำลอง เอาต์พุตของโครงข่ายนั้นถูก Upsample

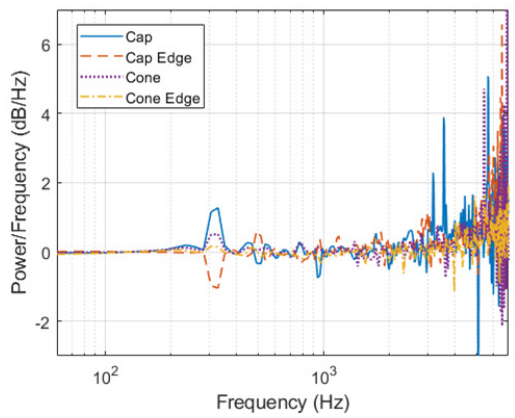


(ก)

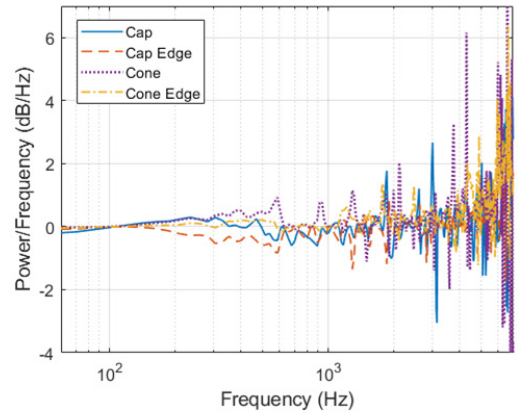


(ข)

รูปที่ 7 ค่า Cross-correlation ของ IR สังเคราะห์เมื่อตั้งค่าไมโครโฟนเป็น (ก) Sennheiser MD421 และ (ข) Shure SM57



(ก)



(ข)

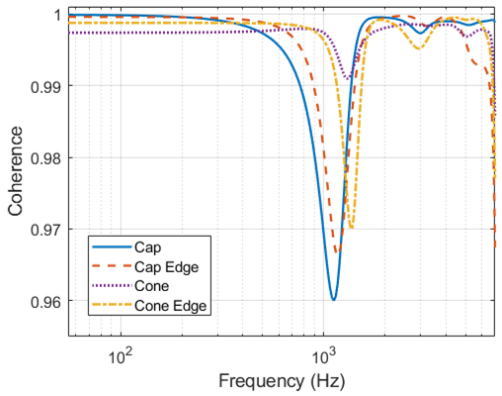
รูปที่ 8 ค่า PSDE ของ IR สังเคราะห์เมื่อตั้งค่าไมโครโฟนเป็น (ก) Royer R121 และ (ข) Sennheiser MD421

ให้ Fs เท่ากันกับ IR ในชุดข้อมูล ตารางที่ 2 แสดงค่าเฉลี่ย MSE, ESR, Cross-correlation สูงสุด (Max Xcorr), PSDE และ MSC ของการทุกรูปแบบการติดตั้งไมโครโฟน เมื่อเปรียบเทียบกับงานวิจัยที่ใช้ ESR เป็นตัวชี้วัด [4] ซึ่งได้ค่าในช่วง 0.2% ถึง 4.2% โมเดลในงานวิจัยนี้มีค่า ESR เฉลี่ยที่ 4.03% ซึ่งต่ำกว่าค่า ESR ของงานวิจัยดังกล่าวเล็กน้อย ตัวอย่างของการวัดค่า Cross-correlation แสดงในรูปที่ 7 โดยที่ค่าสูงสุดที่ lag 0 มีค่าใกล้เคียงกับ 1 ซึ่งหมายถึงสัญญาณมีความใกล้เคียงกัน

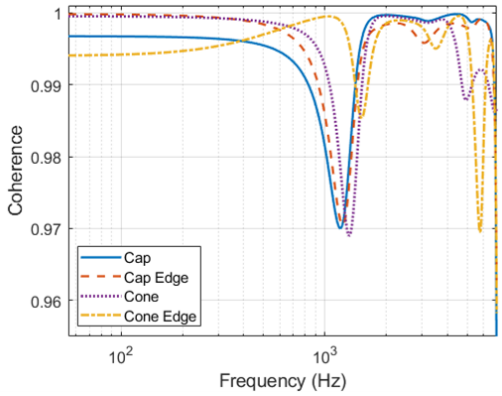
ตารางที่ 2 ผลการทดสอบของแต่ละความถี่ Fs

Fs (Hz)	MSE	ESR	Max Xcorr	PSDE	MSC
44.1k	4.24e-4	5.87%	0.984	3.25e-3	0.880
48k	2.05e-4	2.88%	0.988	3.00e-3	0.930
88.2k	2.55e-4	3.57%	0.997	1.65e-3	0.961
96k	2.71e-4	3.79%	0.998	1.50e-3	0.927

สำหรับอัตราความผิดพลาดของสเปกตรัม รูปที่ 8 (ก) และ (ข) แสดงค่า PSDE ของ IR กำหนดไมโครโฟนเป็น Royer R121 และ Sennheiser MD421 จะสังเกตได้ว่า



(ก)



(ข)

รูปที่ 9 ค่า MSC ของ IR สังเคราะห์เมื่อตั้งค่าไมโครโฟนเป็น (ก) Shure SM57 และ (ข) Neumann KM84

ค่า PSDE นั้นใกล้เคียงหรือเท่ากับ 0 เดซิเบลต่อเฮิรตซ์จากความถี่ต่ำสุดจนถึงประมาณ 1 กิโลเฮิรตซ์ อย่างไรก็ตาม ตั้งแต่ 1 กิโลเฮิรตซ์ ถึงประมาณ 7 กิโลเฮิรตซ์ PSDE จะมีความผิดพลาดเพิ่มขึ้นแต่จะไม่ได้ส่งผลกระทบต่อคุณภาพเสียงอย่างมีนัยสำคัญ เนื่องจากความถี่มูลฐานสูงสุดของกีตาร์มักจะอยู่ในช่วง 1.11–1.32 กิโลเฮิรตซ์ ซึ่งหากจะเกิดผลกระทบเกิดต่อเมื่อมีการใช้เอฟเฟกต์เสียง Distortion (เสียงแตกพร่า) ที่ปรับค่า Gain สูงซึ่งจะสร้างฮาร์มอนิกจำนวนมากเพิ่มขึ้นมาในส่วนของความสอดคล้องของผลตอบสนองความถี่ โดยส่วนใหญ่ ค่า MSC จะใกล้เคียงกับ 1 ในขณะที่มีบางความถี่ เช่นช่วง 1–1.3 กิโลเฮิรตซ์ โดยที่ค่าลดลงเหลือประมาณ 0.96 ดังแสดงในรูปที่ 9 (ก) และ (ข) ซึ่งเป็นค่า MSC ของการติดตั้งไมโครโฟน Shure SM57 และ Neumann KM84

3.2 ประสิทธิภาพในการคำนวณ

สำหรับการตรวจสอบประสิทธิภาพด้านการประมวลผลเสียงตามเวลาจริง จะวัดจากระยะเวลาการคำนวณของปลั๊กอินอันเกิดจากการนำค่าน้ำหนักและไบแอสของโครงข่ายที่ถูกฝึกสอนมาสร้างปลั๊กอินเสียง ทำงานโดยใช้โครงข่ายคำนวณ Forward Pass เพื่อสร้าง IR จากการรับอินพุตจากผู้ใช้และนำ IR ไปทำคอนโวลูชันแบบเร็วกับสัญญาณ มีการใช้ 3 เกณฑ์ เพื่อพิสูจน์ว่าโมเดลสามารถทำงานได้อย่างราบรื่น เกณฑ์แรก คือ Real-time Factor (RTF) [14] สามารถคำนวณได้ตามสมการที่ (12) ยิ่งค่า RTF สูงจะส่งผลให้การประมวลผลแบบเวลาจริงราบรื่นขึ้น

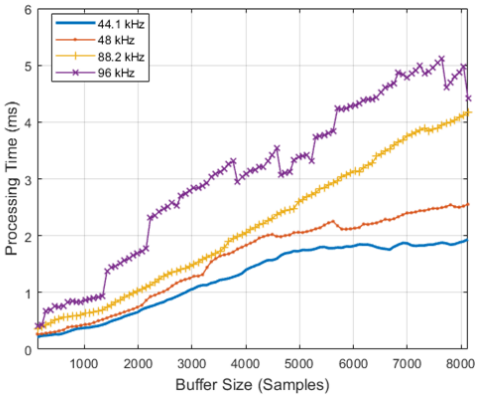
$$RTF = \frac{N \cdot Fs}{t} \tag{12}$$

โดยที่ N คือ ขนาดบัฟเฟอร์หรือบล็อกของเสียงที่ถูกประมวลผล และ t หมายถึงเวลาที่โมเดลใช้ในการประมวลผลสัญญาณแต่ละบล็อก ตารางที่ 3 แสดงค่า RTF ของขนาดบัฟเฟอร์ต่าง ๆ ตั้งแต่ 128 ถึง 8,129 จุดตัวอย่าง ที่ Fs 4 ความถี่ พบว่า RTF ที่น้อยที่สุด คือ 3,431.72 เมื่อ Fs เป็น 44.1 กิโลเฮิรตซ์ และขนาดบัฟเฟอร์ คือ 128 จุดตัวอย่าง ส่วนค่า RTF สูงสุด คือ 214,585.27 ซึ่งจะปรากฏขึ้นเมื่อ Fs เท่ากับ 96 กิโลเฮิรตซ์ ด้วยขนาดบล็อก 8,129 จุดตัวอย่าง

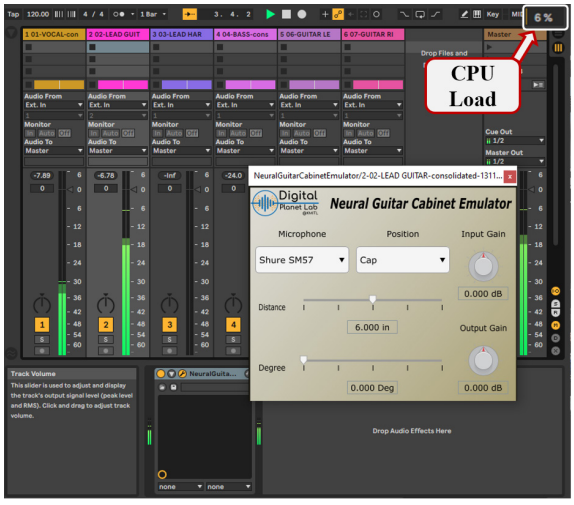
ตารางที่ 3 ผลการทดสอบของแต่ละความถี่ Fs

Fs	Buffer Size (Samples)						
(Hz)	128	256	512	1024	2048	4096	8129
44.1k	34.3e3	68.2e3	13.4e4	26.5e4	52.3e4	10.3e4	20.4e4
48k	35.7e3	70.8e3	13.9e4	27.1e4	53.3e4	10.5e4	20.7e4
88.2k	35.7e3	70.3e3	13.8e4	27.2e4	53.6e4	10.6e4	20.9e4
96k	36.7e3	72.1e3	14.2e4	27.9e4	55.1e4	10.9e4	21.5e4

เพื่อวัดประสิทธิภาพการคำนวณเพิ่มเติมซอฟต์แวร์ Plugin Doctor ถูกนำมาใช้ ฟังก์ชันวัดประสิทธิภาพของ

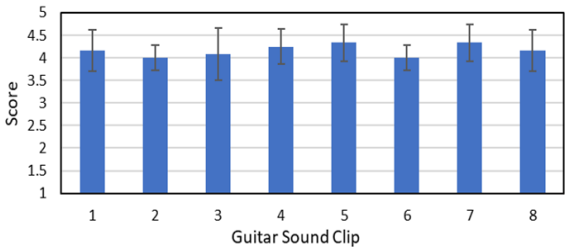


รูปที่ 10 เวลาประมวลผลของโมเดลที่ Fs แต่ละความถี่



รูปที่ 11 GUI ของปลั๊กอินและโหลด CPU ที่เปิดใน DAW (Ableton Live)

ซอฟต์แวร์แสดงในรูปที่ 10 โดยมีเวลาในการประมวลผลต่อบล็อกที่ช้าที่สุด 5.12 มิลลิวินาที เมื่อตั้งค่า Fs เป็น 96 กิโลเฮิรตซ์ และขนาดบัฟเฟอร์ตั้งไว้ที่ 8129 จุดตัวอย่าง การตั้งค่าที่ Fs เป็น 44.1 กิโลเฮิรตซ์ และขนาดบัฟเฟอร์ 128 จุดตัวอย่างต่อบล็อก ใช้เวลาเพียง 0.22 มิลลิวินาที ในการประมวลผลแต่ละบล็อกซึ่งเร็วที่สุดในบรรดาบล็อกอื่น ๆ เกณฑ์สุดท้ายคือมิเตอร์วัดการใช้งาน CPU ใน DAW ที่ชื่อ Ableton Live รูปที่ 11 แสดงภาพหน้าจอของ Ableton Live ที่โฮสต์ของปลั๊กอินของโมเดลที่เสนอและยังค่าโหลด CPU ที่ใช้ผลลัพธ์แสดงให้เห็นว่าโหลด CPU เฉลี่ยสำหรับการ



รูปที่ 12 คะแนนความคล้ายคลึงกันของการทดสอบการฟัง

ประมวลผลเสียงใน DAW คือ 6% และในระหว่างการปรับพารามิเตอร์ โหลดจะอยู่ที่ 9% ถึง 34% ผลลัพธ์เหล่านี้ชี้ให้เห็นว่าปลั๊กอินนี้มีประสิทธิภาพในการคำนวณ และสามารถประมวลผลแบบเวลาจริงได้

3.3 การทดสอบฟัง

ในการวัดด้านการรับรู้ของคุณภาพเสียงด้วยการฟัง ได้มีการทดสอบการฟังแบบให้คะแนนความคิดเห็นเฉลี่ย หรือ MOS (Mean Opinion Score) วัตถุประสงค์ คือ ตรวจสอบความเหมือนของเสียงกีตาร์ที่ผ่านการจำลองกับ IR ที่บันทึกจริงและ IR ที่สร้างโดยโครงข่าย คลิปเสียงกีต้าร์มีทั้งหมด 8 คลิป แต่ละคลิปมี 3 รูปแบบ ได้แก่ สัญญาณ dry (ไม่ผ่านการใส่เอฟเฟกต์หรือการจำลองลำโพงใด ๆ) สัญญาณที่ผ่านการคอนโวลูชันกับ IR ที่บันทึกจริง และสัญญาณกีต้าร์ที่ผ่านการคอนโวลูชันกับ IR ที่สร้างโดยประสาทเทียม ชุดทดสอบมีทั้งเสียงกีตาร์ที่บันทึกด้วย DI Box แบบ Clean และแบบที่ใช้เอฟเฟกต์ Distortion ทุกคลิปมีความยาว 2 วินาที คะแนนมีตั้งแต่ 1 ถึง 5 โดยคะแนน 1 หมายถึงเสียงต่างกันโดยสิ้นเชิง และคะแนน 5 หมายถึงเสียงทั้งสองฟังเหมือนเป็นเสียงเดียวกัน

มีผู้เข้าร่วมทำแบบทดสอบรูปแบบออนไลน์ 12 คน ซึ่งทุกคนมีประสบการณ์ที่เกี่ยวข้องกับงานด้านดนตรี ผู้เข้าร่วมได้รับคำแนะนำให้ฟังผ่านหูฟังหรือลำโพงมอนิเตอร์ และอนุญาตให้เล่นคลิปเสียงซ้ำได้ตามต้องการ ผลการทดสอบ MOS แสดงในรูปที่ 12 คะแนนเฉลี่ย คือ 4.167 ± 0.130 โดยมีช่วงความเชื่อมั่น 95% ซึ่งถือว่าใกล้เคียงกับเสียงต้นฉบับอย่างชัดเจน

4. อภิปรายผลและสรุป

งานวิจัยได้มีการออกแบบโมเดล MLP ที่มี 5 ชั้นซ่อน และใช้เพื่อสังเคราะห์สัญญาณ IR ของลำโพง Marshall Guitar Cabinet โครงข่ายรับ 4 อินพุต ได้แก่ ประเภทไมโครโฟน ตำแหน่งดอกลำโพงที่ตั้งไมโครโฟนหันเข้าหา ระยะห่าง และมุมเอียง และให้เอาต์พุตเป็น IR ความยาว 20.5 มิลลิวินาที IR ที่โครงข่ายทำนายถูกประเมินในโดเมนเวลาและความถี่ รวมไปถึงทดสอบด้วยการฟัง ผลการทดลองแสดงว่าเสียงการจำลองด้วย IR สังเคราะห์มีความใกล้เคียงกับการจำลองด้วย IR จริงทั้งในแง่ของการฟังและการประเมินด้วยเกณฑ์ต่าง ๆ สำหรับประสิทธิภาพการคำนวณ ได้มีการทดลองสร้างโมเดลโครงข่ายปราสาทเทียม CNN และ RNN แบบ LSTM จากงานวิจัยที่ [4] และพบว่า ใช้โหลด CPU ตั้งแต่ 40–70% และใช้เวลาประมวลผลสูงสุดถึง 200 มิลลิวินาที ซึ่งนับว่าช้าการโมเดล MLP ที่ใช้โหลด CPU ประมาณ 6% ใช้เวลาประมวลผลไม่เกิน 6 มิลลิวินาที เพราะมีความซับซ้อนที่น้อยกว่า โมเดลที่เสนอจึงสามารถนำไปประยุกต์เป็นปลั๊กอินเสียงใช้งานแบบเวลาจริงได้

ในกระบวนการปรับโทนกีตาร์ ผู้ใช้จะต้องปรับแต่งการตั้งค่า IR ของโพงไปมาเพื่อให้ได้ยินความแตกต่างและหาแบบที่เหมาะสมที่สุด โดยทั่วไปจำเป็นต้องมีไฟล์ IR จำนวนมาก พร้อมทั้งรูปแบบการตั้งไมโครโฟนที่หลากหลายซึ่งจะต้องใช้ปลั๊กอิน IR loader โหลดทีละไฟล์ การทำงานรูปแบบดังกล่าวมักส่งผลให้ผู้ใช้ลืมเสียงการจำลองกับ IR ก่อนหน้าเมื่อถึงเวลาที่มีการโหลดไฟล์ IR ใหม่ โมเดลที่นำเสนอช่วยลดปัญหาความไม่ต่อเนื่องในการทำงานนี้ด้วยการสังเคราะห์ IR ตามการตั้งค่าของผู้ใช้แบบเวลาจริง ทำให้ผู้ใช้สามารถฟังความแตกต่างของเสียงจำลองระหว่างการปรับรูปแบบการตั้งไมโครโฟนต่าง ๆ ไปมาได้ และยังสามารถสังเคราะห์ IR ของการติดตั้งไมโครโฟนลักษณะอื่นที่ไม่รวมอยู่ในชุดการฝึกได้ด้วยซึ่งเกิดจากการเรียนรู้ความสัมพันธ์ของข้อมูลในชุด นอกจากนี้จะช่วยลดความต้องการพื้นที่หน่วยความจำสำหรับการในจัดเก็บไฟล์ IR จำนวนมาก การใช้โมเดลนี้ในงานด้านอุตสาหกรรมดนตรีจึงช่วยเพิ่มความสะดวกยิ่งขึ้น

ทั้งนี้ยังสามารถเพิ่มความแม่นยำได้โดยบันทึกเสียงเพิ่มสำหรับชุดข้อมูล รวมไปถึงใช้โครงข่ายแบบอื่นด้วย ซึ่งทั้งหมดนี้อาจเป็นแนวทางการวิจัยในอนาคตเพื่อให้ได้ผลที่ดียิ่งขึ้น

เอกสารอ้างอิง

[1] A. Ratnarajah, Z. Tang, and D. Manocha, “IR-GAN: Room impulse response generator for far-field speech recognition,” in *Interspeech 2021*, 2021.

[2] N. J. Bryan, “Impulse response data augmentation and deep neural networks for blind room acoustic parameter estimation,” in *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[3] C. Steinmetz. (2021, October). NeuralReverberator [Online]. Available: https://www.christiansteinmetz.com/projects-blog/neural reverberator

[4] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, “Real-time guitar amplifier emulation with deep learning,” *Applied Sciences (Basel)*, vol. 10, no. 3, pp. 766, 2020.

[5] M. A. Martinez Ramirez and J. D. Reiss, “Modeling nonlinear audio effects with end-to-end deep neural networks,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[6] M. A. Martínez Ramírez, E. Benetos, and J. D. Reiss, “Deep learning for black-box modeling of audio effects,” *Applied Sciences (Basel)*, vol. 10, no. 2, pp. 638, 2020.

[7] G. Stan, J. Embrechts, and D. Archambeau, “Comparison of different impulse response measurement techniques,” *Journal of the*

Audio Engineering Society, vol. 50, no. 4, pp. 249–262, 2002.

[8] S. W. Smith, *The scientist and engineer's guide to digital signal processing*. California Technical Pub, 1997.

[9] A. Ratnarajah, S.-X. Zhang, M. Yu, Z. Tang, D. Manocha, and D. Yu, "FAST-RIR: Fast neural diffuse room impulse response generator," *arXiv [cs.SD]*, 2021.

[10] A. Wright and V. Valimaki, "Perceptual loss function for neural modeling of audio systems," in *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[11] Redwirez (2021, October). *FREE Marshall 1960 IRs* [Online]. Available: https://redwirez.com/pages/the-marshall-1960a-ir-pack.

[12] Suhr.com. (2021, October). *PT 15 IR User Guide* [Online]. Available: https://www.suhr.com/wp-content/uploads/2020/10/PT-15-IR-User-Guide-100120.pdf.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv [cs.LG]*, 2014.

[14] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time analog audio effect modeling," *arXiv [eess.AS]*, 2021.