# Machine Learning for the Inverse Control of FM Synthesis

Rosa Garza[1], Andrew Pfalz[2], and Dr. Edgar Berdahl[2]
*California State University, Monterey Bay[1],*
*Center for Computational & Technology, Louisiana State University[2]*

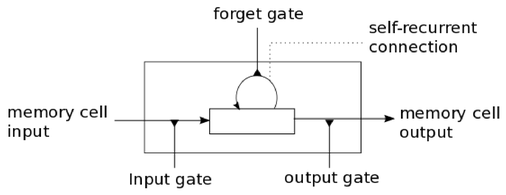
## Introduction

Frequency Modulation (FM) is an efficient sound design procedure where control signals are input to a synthesizer and audio is output. This technique is often used in FM radio. However, to reproduce a particular sound, the exact control signals are needed for the FM synthesizer. If the control signals are not known, there will be a repetitive process of changing the control signals and listening to the audio until the sound closely matches.

A possible solution to this situation is to teach a recurrent neural network (RNN) to learn the control signals and to mimic how they modify audio. Our research explores the area of using a Long Short Term Memory (LSTM) RNN model to learn what control signals were inputted in the FM synthesizer. This is the development towards an FM inverse synthesizer. With an FM inverse synthesizer, an LSTM RNN can receive audio and output the control signals used in the FM synthesizer.

## LSTM Neural Network Model

An LSTM model was used rather than other RNN's because of the model's design for sequence learning. The LSTM RNN's inner architecture include a memory cell (see Figure below)[1] which helps to eliminate the issues that can arise in back-propagation.




## Methods

**Calculating loss:** Mean Squared Error
- We interpret as the difference between the correct audio label and the LSTM's answer.
- **Goal:** For the loss to be a very small number close to 0.

**LSTM Hyperparameters adjusted:** Learning rate, number of unrollings, epochs, and number of LSTM layers

**Control signals:** Carrier frequency (cf), depth (d), & modulation frequency (mf)
- Time varying
- Randomized float between [0,1) and normalized

**Lengths of audio input:** 0.1 sec., 1 sec., 10 sec., 30 sec., 1 min.

**Creating Audio:** Control signals were input into the FM synthesizer through the following equation:

$$audio = \sin(2\pi \cdot t \cdot cf + \left(\frac{d}{m}\right)\sin(2\pi \cdot t \cdot mf))$$

*t is an array of time

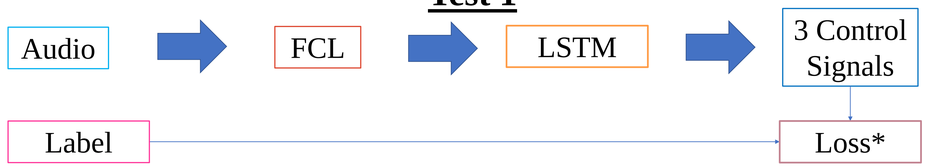
- Fourier Transform (FT) was applied to audio for LSTM to receive information on data of frequencies and each frequency's power throughout the audio.

**Label for the audio:** Concatenation of the three control signal arrays [cf:d:mf]

## Results

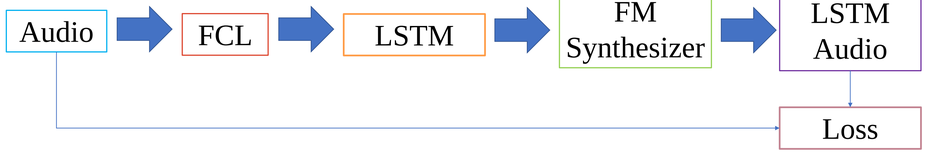
Below are the different architectures for the process of testing on the audio data. All architectures were developed in order to continue making improvements and outputting audio similar to the original input audio. FCL, in the figures below, stands for " fully connected layer".

### Test 1

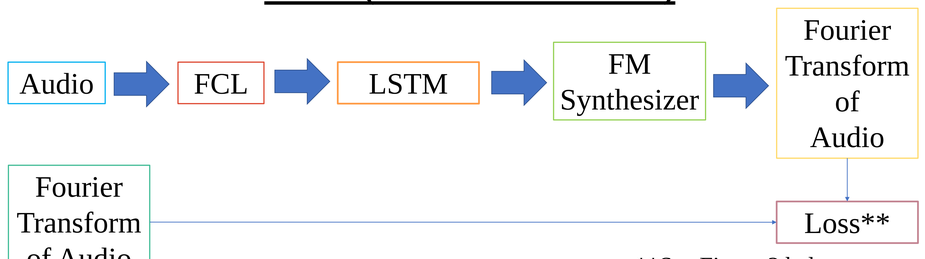



*See Figure 1 below

### Test 2




### Test 2 (Fourier Transform)




**See Figure 2 below

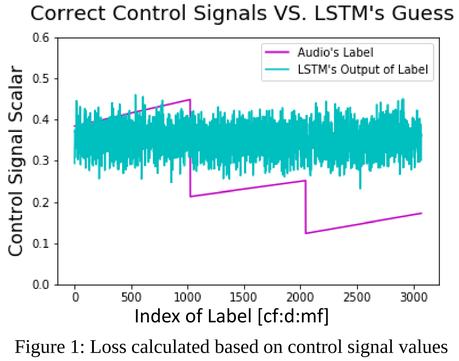



Figure 1: Loss calculated based on control signal values

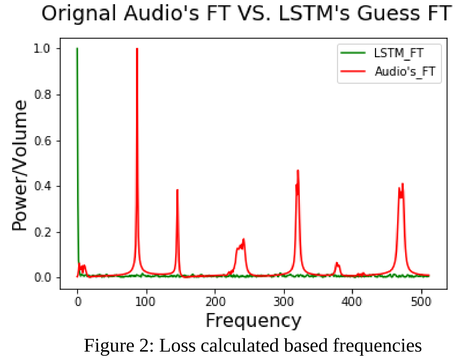
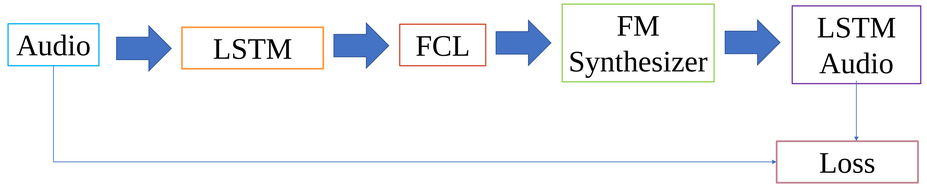
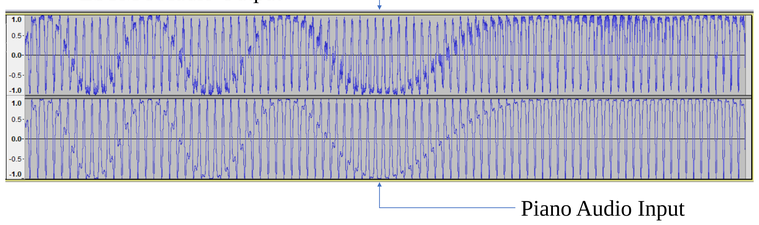



Figure 2: Loss calculated based frequencies

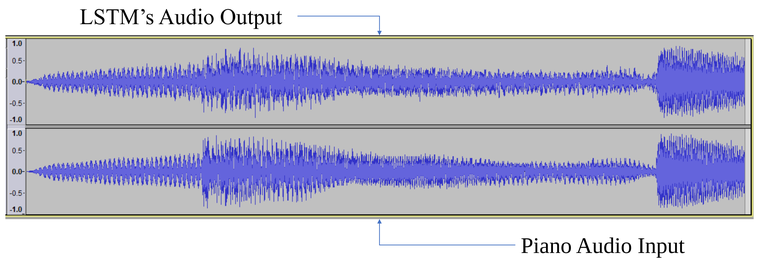
### Test 3




Small Audio Input




Piano Input




## Discussion

From the results, it appears that the order of applying the fully connected layer makes a huge difference in the LSTM's performance. When applying the fully connected layers before the LSTM RNN, the Inverse Control of FM Synthesis does not output audio close in sound to the original audio input.

The fully connected layer is then applied right after the LSTM's output for Test 3. From this architecture, the LSTM's audio output has been shown and produces much more promising results. Overall it shows that it is the best architecture for the Inverse Control of FM Synthesis. Not only is the LSTM RNN able to receive a randomly generated audio sample, but it is also able to receive a piano audio and output a very similar audio with a small amount of noise.

## Conclusion

The last architecture for the Inverse Control of FM synthesis is a very successful procedure. For training on audio, it is shown that inputting the raw audio data into the LSTM and then applying fully connected layers is the best approach. For future work, it will be best to try and apply the Fourier Transform. From past experiments, applying the Fourier Transform did perform better than just training on raw audio. This technique may help the Inverse Control of FM Synthesis to adjust the output to having the same frequency and reduce noise.

## Acknowledgements

I would like to thank my graduate student, Andrew Pflaz, Dr. Edgar Berdahl, and Dr. Jesse Allison . Also, I would like to thank my family and friends for their support throughout my research experience. This research was supported by the National Science Foundation under award OCI-1560410 with additional support from the Center for Computation & Technology at Louisiana State University. Additional computer support was also provided by Titan@CCT.

## References

1. LSTM Networks for Sentiment Analysis: http://deeplearning.net/tutorial/lstm.html