Tampere University

# Sound synthesis and manipulation

SGN 14007

Lecture 5

Annamaria Mesaros

# Sound synthesis and its uses

Sound synthesis

- The goal is to produce sounds that are musically interesting and (if required) resemble a real instrument
- Sounds have to be produced in real time
- Intuitive control of the synthesized sound is important: interaction of the player makes the sound lively

Audio effects include for example

- Dynamic level control (discussed on a separate lecture)
- Equalization (discussed on a separate lecture)
- Time stretching and pitch shifting
- Adding reverberation (in its simplest form, simply filtering the ”dry” sound with the impulse response of the target reverberant room)
- Effects applied on musical sounds: flanger, chorus, phaser, wah-wah

# Additive synthesis

Idea:

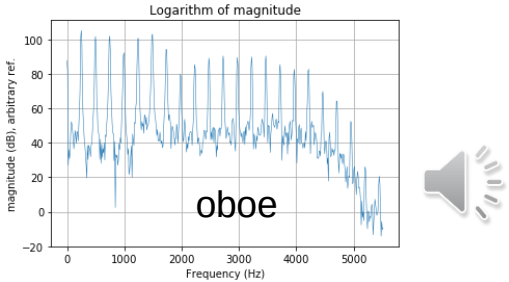
- Any (almost) periodic sound can be represented as a sum of sinusoids.
- Each sinusoid's frequency is a nearly multiple of the fundamental frequency

Generally:

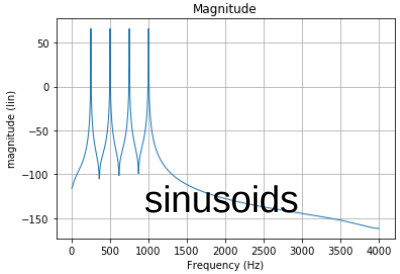
$$y(t) = \sum_k A_k(t) \sin\left[2\pi f_k(t) t\right]$$

- The amplitudes and frequencies $A_k$ and $f_k$ of individual sinusoids are slowly time-varying, e.g., depend on t

Example:






# Additive synthesis

In principle enables very high quality

Drawbacks:

- Requires a lot of data (time-varying parameters) for each note of each instrument
- Large number of oscillators

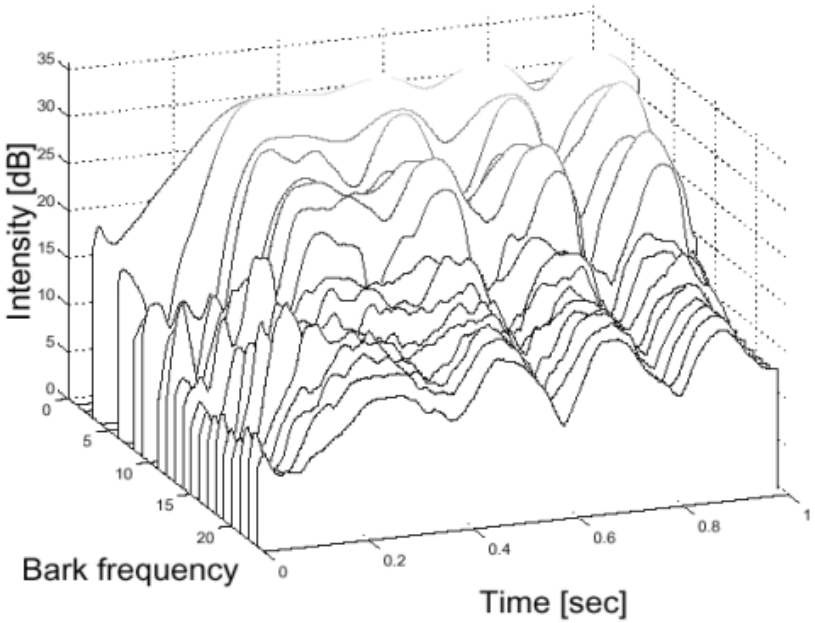



Figure: temporal evolution of the harmonic frequencies of a flute sound

# FM synthesis

# FM synthesis

- Frequency modulation (FM) has been used in telecommunications, for example in broadcast radio, already for more than half a century

- In late 1960s, John Chowning proposed to apply FM-synthesis for sound production
  - Idea: complex spectra can be synthesized with only a few voltage controlled oscillators
- In 1983 Yamaha released its DX7 synthesizer
  - Great commercial success: an instrument that had high sound quality and price that could be afforded by the consumers
- FM synthesis stayed as the dominant synthesis method for years

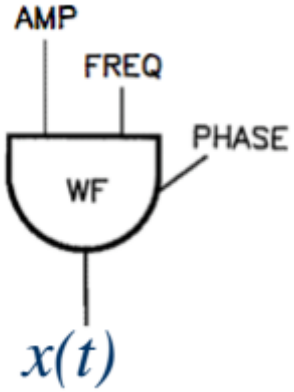
- Implementing FM-synthesis in its simplest form requires only few lines of code and a few parameter values

# FM synthesis

- Oscillator (basic building block in FM synthesis)

  $x(t) = A(t) \sin 2\pi f t + \varphi$

  - frequency f
  - (Time-varying) amplitude A(t)
  - $\varphi$ is phase (rad)



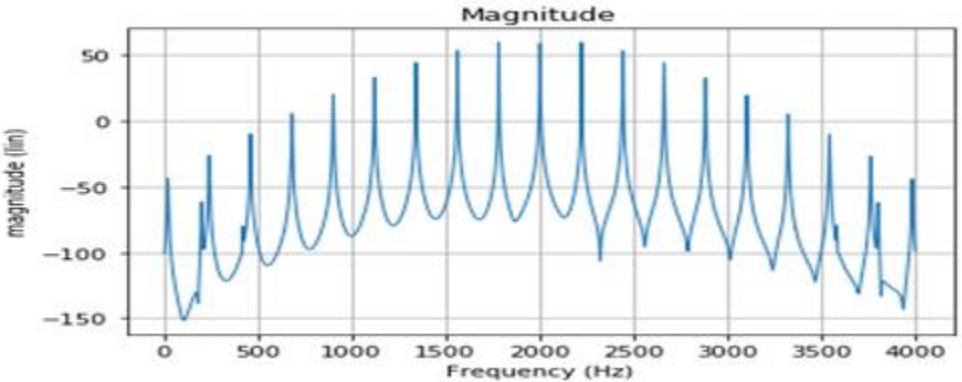

- Modulation:
  - Varying of frequency, amplitude, or phase of the oscillator as a function of time
  - Used also in radio-communication
  - Used also in creation of different effects in electronic music

# FM synthesis

Simple FM synthesis

$$y(t) = A(t) \cdot sin\left(2\pi f_c t + I \cdot sin(2\pi f_m t)\right)$$

- Consists of two sinusoidal oscillators
- Carrier frequency $f_c$ is modulated by another oscillator (modulation frequency $f_m$)
- Modulation index I
- Time-varying amplitude A(t)



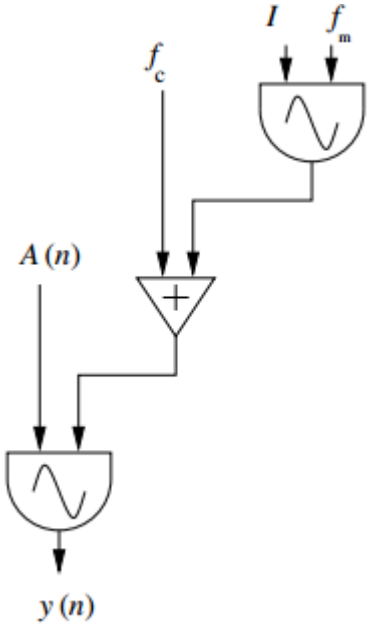




Figure: spectrum of the resulting sound, $f_c$=2000Hz,$f_m$=220Hz, I=0.5

# FM synthesis

$$y(t) = A(t) \cdot sin\left(2\pi f_c t + I \cdot sin(2\pi f_m t)\right)$$

- Note that we modulate phase instead of frequency!
  - more correct name would be phase modulation (PM)
- PM and FM produce essentially the same kind of sound

- In analog devices, FM is used: PM is practical only digitally
- The above equation does not express the relationship of the magnitude of harmonic frequencies.
- sin(a + b sin(c)) can be written as a sum of weighted sinusoids using Bessel functions

For those who want more details, not important for exam

# FM synthesis

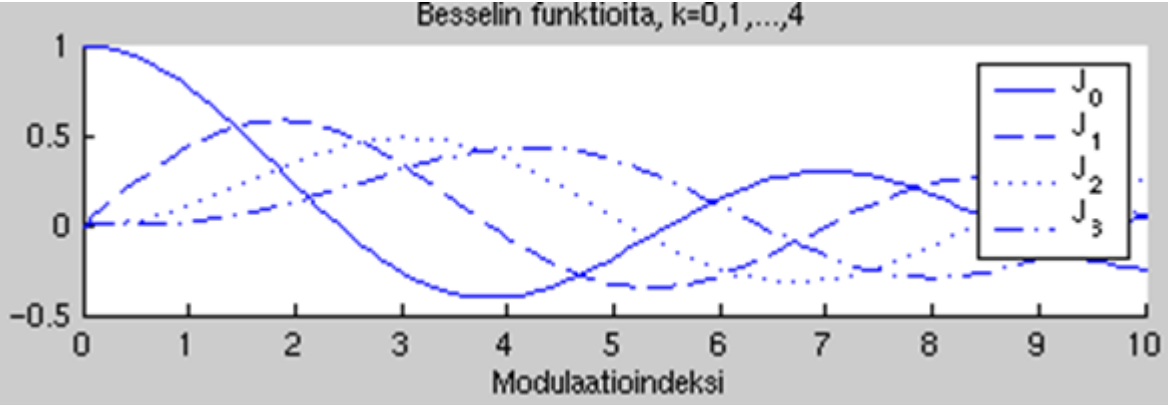
The formula on the previous slide can be written as $x(n) = A(n)\sum_{k=-\infty}^{\infty} J_k(I)\sin\left[2\pi(f_c + kf_m)n\right]$

where $J_k$ is a Bessel function (1st kind) of order *k* evaluated at I.

- From the above formula we can see that PM synthesis (and FM synthesis) produces frequency components

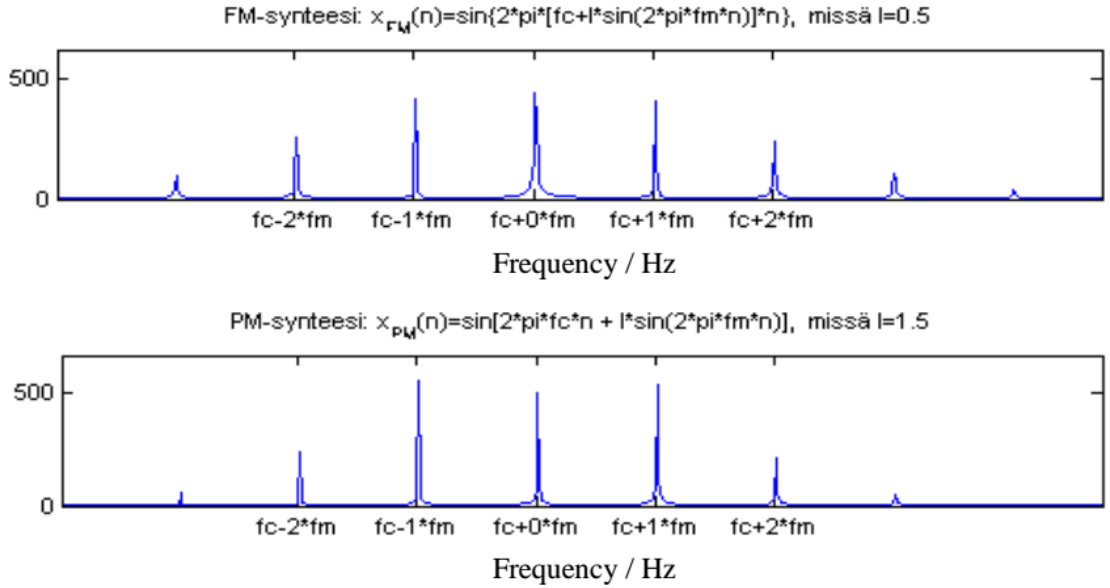
$$f_n = f_c \pm kf_m, k = 1,2,\ldots$$

Figure: Bessel functions Jk(I) for k = 0,1,…,3



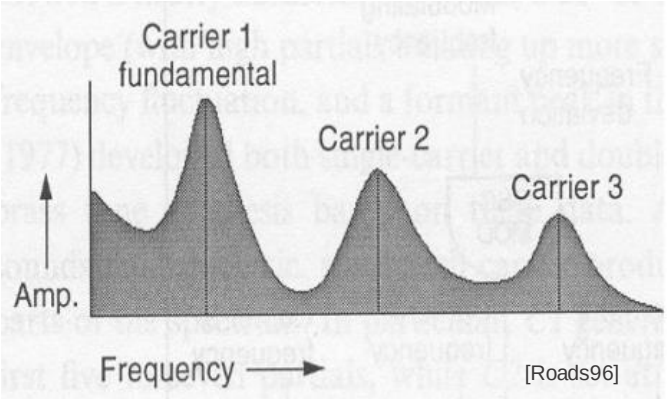
# FM synthesis

- Harmonic sound is obtained by setting the carrier and modulation frequencies into a whole-number ratio
- FM vs PM:



# FM synthesis: extensions

- Compose the desired spectrum by summing the output of several basic FM blocks




[Roads96]

# Sampling synthesis

# Sampling synthesis

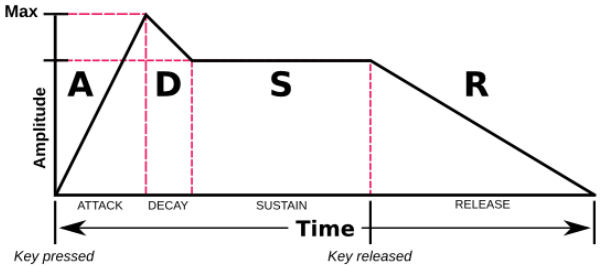
- Recordings of relatively short sounds are played back
  - Samplers: digital sampling instruments: perform pitch shifting, looping, other modifications to the original signal
- Very widely-used synthesis method

- Memory requirements are the challenge of this approach
  - It is not economical to store all possible sounds from all instruments
  - The art of sampling synthesis is to minimize the memory consumption while maximizing the sound quality
  - Basic idea: to model sounds with less data

# Looping

- For most musical instruments, the steady-state (= sustain) part is nearly periodic
  - Can take a short sample and play that in a loop buffer
  - The short beginning transient of a sound can be modeled separately

- Looping part should be long enough to make it sound natural
  - Short loop sound clearly periodic
  - Even a very long loop repeated enough times reveals its periodicity
- Temporal model of a note: Attack, Decay, Sustain, Release (ADSR)
  - Loop extracted from the sustain part (between looping points)
- Enveloping: change perception of a sample by modifying ADSR curve to speed-up decay, shorten the attack time



# Multiple wavetable synthesis

- Several buffers (wavetables) are played back simultaneously

- Wavetable cross-fading
  - Store samples of the sound at several temporal positions
  - Cross-fade smoothly from one loop buffer to the next
  - Windowing can be applied, make the windows overlap + sum to 1

- Wavetable stacking
  - The desired waveform is constructed by forming a weighted sum of several elementary waveforms that are played simultaneously (additive synthesis!)
  - Challenge: find a set of elementary waveforms (and amplitude envelopes) to represent various natural sounds

# Phase vocoder

# Phase vocoder

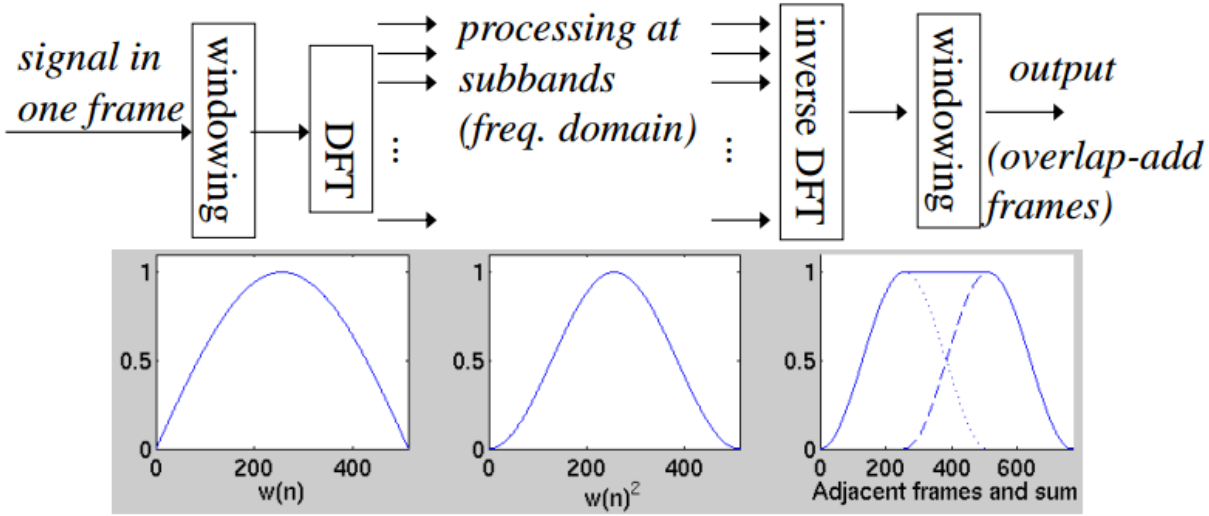
- ”vocoder” = **vo**ice **coder**: invented in the 60s when researching speech compression methods
  - Present the signal in multiple parallel channels, each describing the signal in a particular frequency band →simplify the spectral information, reduce the amount of data needed to present the signal
- Applications nowadays: audio time stretching, pitch shifting, audio morphing, time-frequency domain processing

- General name for analysis-synthesis methods where
  - Audio signal is represented as a **sum of sinusoids**
  - In addition to magnitudes and frequencies, also **phases are synthesized**

- Usually implemented using short-time Fourier transform (STFT)
- Spectrum modification method: allowing modifications to the amplitudes or phases of specific frequency components of the sound, before resynthesis

# Analysis-synthesis systems

- Sine window (left panel below) is useful in analysis-synthesis systems
  - squared sine window equals the hanning window (middle panel)
- Sine windowing is done again in resynthesis to avoid artefacts at frame boundaries in the case that signal is manipulated in the f-domain
  - 50% frame overlap leads to perfect reconstruction if nothing is done at subbands



# Time stretching and pitch shifting

- Phase vocoder allows audio time stretching and pitch shifting

- Time stretching
  - Change the time interval between frames during synthesis,
  - OR alternatively, frames are copied or deleted at suitable intervals
  - Time-domain windowing has to be done carefully to avoid artefacts
  - Phases are processed to keep phase time derivative unchanged (more details in exercise this week)

- Pitch shifting
  - A single recording of an instrument can be pitch shifted over a frequency range
  - The range producing credible results depends on the instrument
    - Piano: one note per octave (there are 12 notes in octave) can be sufficient
    - Horn, human voice: several recordings per octave are required

# Time stretching and pitch shifting

- Pitch shifting
  - Time-stretch first, then resample the signal so that duration becomes the same again, but pitch changes
  - Large changes make especially speech sound strange, because the coarse spectral envelope is shifted too (formants)


Low


Original


High

- Using the spectrogram representation, sound can be flexibly processed before returning to the time domain
  - Filtering: multiply with frequency response
  - Morphing sounds: interpolate the magnitude and phase spectra of sounds

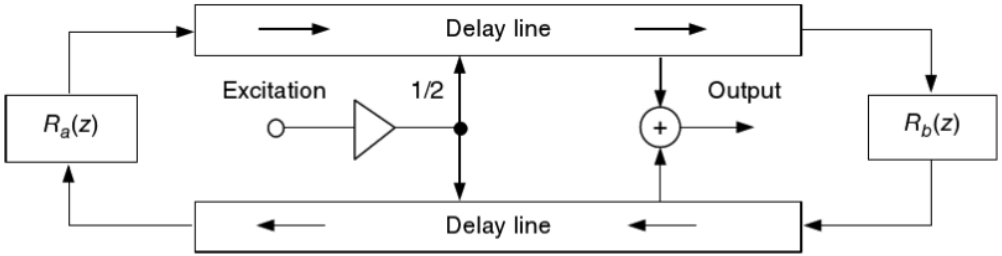
# Physical modeling

- Musical acoustics: physics branch aimed to improve understanding of instruments
- A mathematical model is developed for an instrument
  - Typically a set of differential equations describing components of the instrument
  - The motion of the real system can be modeled accurately given boundary conditions and external forces
  - Numerical methods are used to solve the equations by discretizing them using finite difference approximation methods.
- Challenges
  - Computational complexity
  - Producing realistic sounds
- Benefits
  - Small memory footprint
  - Parametric representation allows to generate wide range of sounds

# Digital waveguides

- Idea: Use digital filters to model the traveling wave.
- Can allow real-time synthesis (computationally efficient physical modeling), with high quality when applicable
  - Linear elements of one-dimensional with uniform mass distribution
  - E.g. vibrating strings, pipes, horns

- Figure: digital waveguide to model a string of a guitar
  $R_a(z)$ and $R_b(z)$ model attenuation of vibration at the terminals



# Neural audio generation

# Neural audio generation

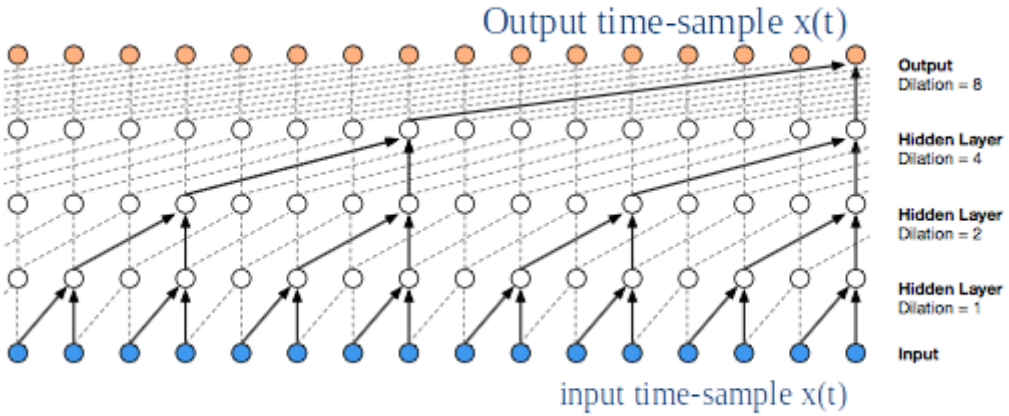
- Modern deep learning approaches present a novel and relatively unexplored way for sound synthesis
- They model the probability density of actual data $p_D(x)$ with a model $p_{model}(x)$.
- A joint-probability model is often applied

$$p(\boldsymbol{x}) = \prod_{t=1}^{T} p(x_t | x_1, x_2, \ldots, x_{t-1}),\ \boldsymbol{x} = [x_1, x_2, \ldots, x_t]$$

- $p_{model}(x)$ needs to be learned from the data
  - Model parameters are iteratively updated so that $p_D(x)$ is modeled accurately by $p_{model}(x)$
- Can be unconditional or conditional to external input
  - Conditioning information can be e.g. desired note: $p_{model}(x|note)$
- Sampling from $p_{model}(x)$ will produce a signal, that resembles an actual signal, sampling from $p_{model}(x|note)$ produces a signal to model the note

# NSynth neural audio synthesis

- Uses a Wavenet as an autoencoder
  - Wavenet is a convolutional neural network, with dilated convolutions to increase the “receptive field” size
  - Can model any kind of audio

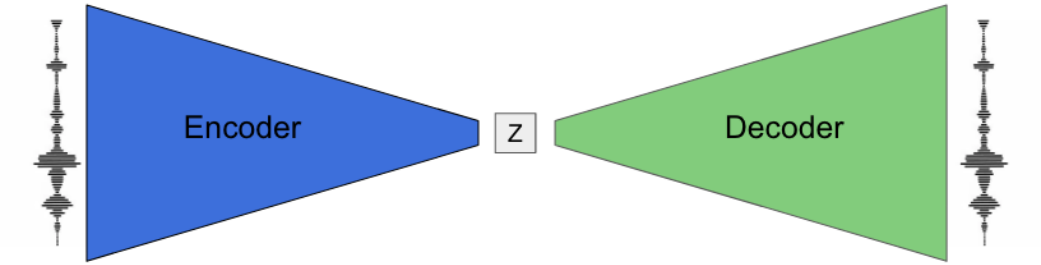


# Wavenet

- An autoregressive model (think of an IIR filter)
- Structure:
  - Contains 30 convolutional layers with dilation: 1,2,4,…,512,1,2,3,…,512,1,2,4,…512
  - Efficient way to model that the current output sample depends on thousands of past samples
  - Uses skip-connections and gated activation units
  - Output value is a class from 1,…,256, where each class represents a u-law compressed sample value
    - u-law is non-linear quantization that applies logarithm for amplitude.
- Training:
  - Learn to predict next sample given past samples
  - Update model parameters based on the prediction error
- Testing
  - Predict sample $x_t$ using T past predicted samples.
  - Place predicted sample to place, predict $x_{t+1}$ using past T-1 samples

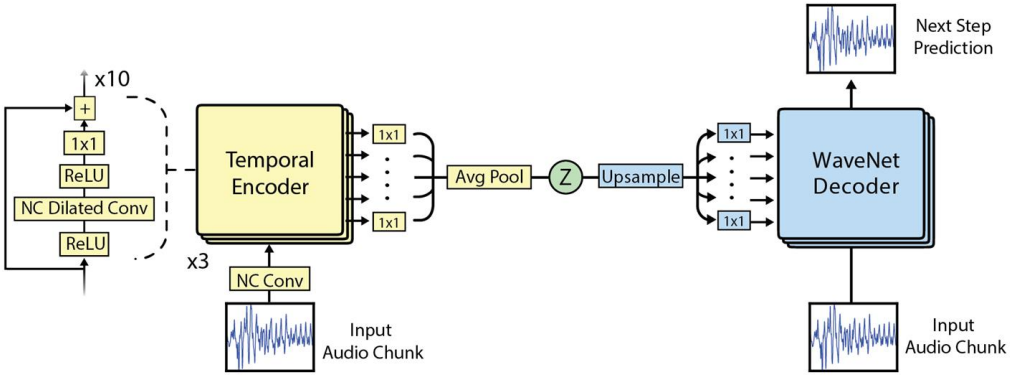
# Autoencoder

- Artificial neural networks with an encoder and a decoder stage
- Encoder represents the input with less dimensions
  - Forces a compressed view / representation of the input ( Z-space in figure)
- Decoder produces the output from the compressed representation
- Output = Input



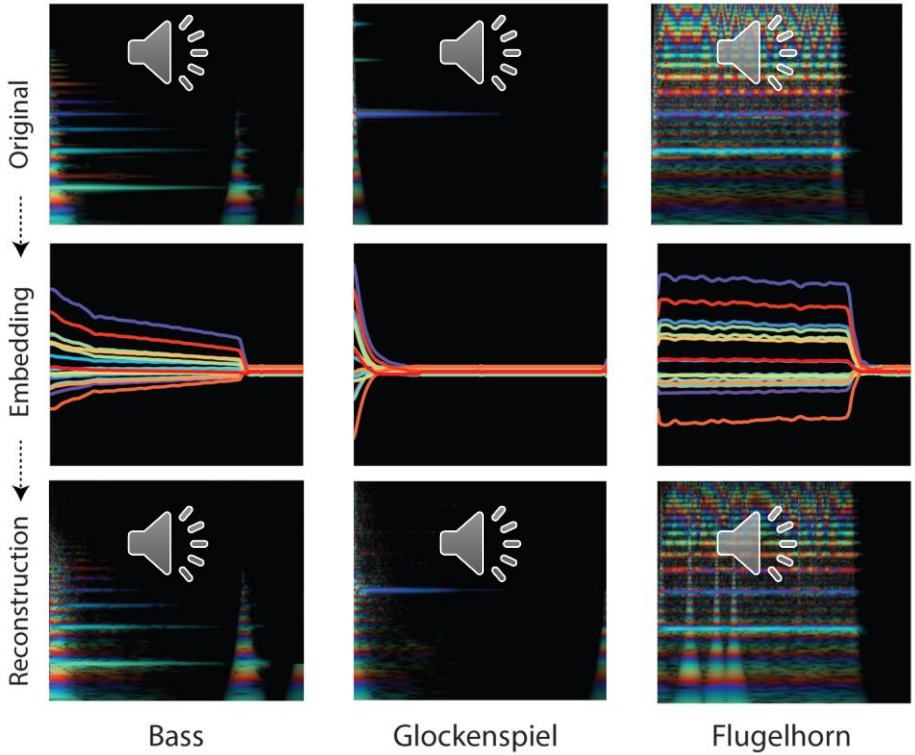
# NSynth

- Temporal encoder: reduces data to Z-space (embedding)




- Wavenet decoder: reconstructs the signal using also the interpolated Z-space representation
- The decoder is conditioned on the Z-space embedding

# Reconstructed samples



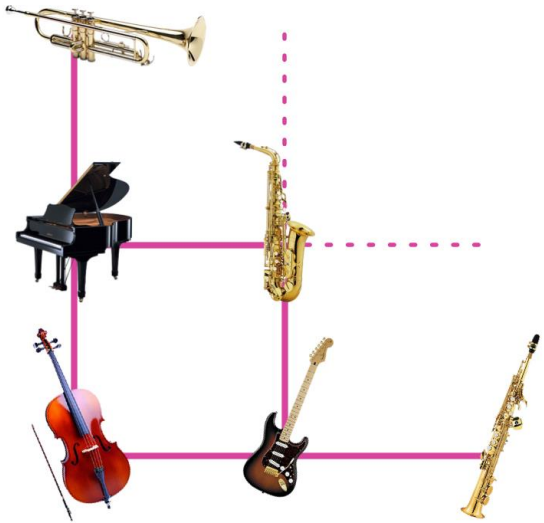

Read more at: https://magenta.tensorflow.org/nsynth

# NSynth

- Z-space embedding allows new hybrid instruments: mixing the encoded signals of two separate sounds
- Time stretching
- Demos:
  - https://www.youtube.com/watch?v=rU2ieu5o5DQ
  - https://experiments.withgoogle.com/ai/sound-maker/view/
- Model training
  - 305979 musical notes, 1006 instruments
  - “The WaveNet model takes around 10 days on 32 K40 gpus…”.
- Synthesis
  - Pre-trained models are available for synthesis [2]

    https://github.com/tensorflow/magenta-demos/tree/master/jupyter-notebooks
  - Samples: time stretching

Original Resynth Fast Slow



# Summary

- Different approaches for sound synthesis:
  - Additive synthesis, FM synthesis
  - Sampling synthesis
  - Physical modeling
  - Neural modeling

- Challenges and benefits of each approach