

Sampling & Sequencing

Combining MIDI and audio

Introduction

- ▶ To create audio assets we often want to combine MIDI data with audio
- ▶ How to combine multiple audio assets to create a single artefact?
- ▶ How to add in MIDI which is **not** an audio format?
 - ⦿ Need to convert synth output to audio
- ▶ We will use a special program called a sequencer
 - ⦿ Like a multi-track tape recorder it allows us to build up songs/tracks from component parts

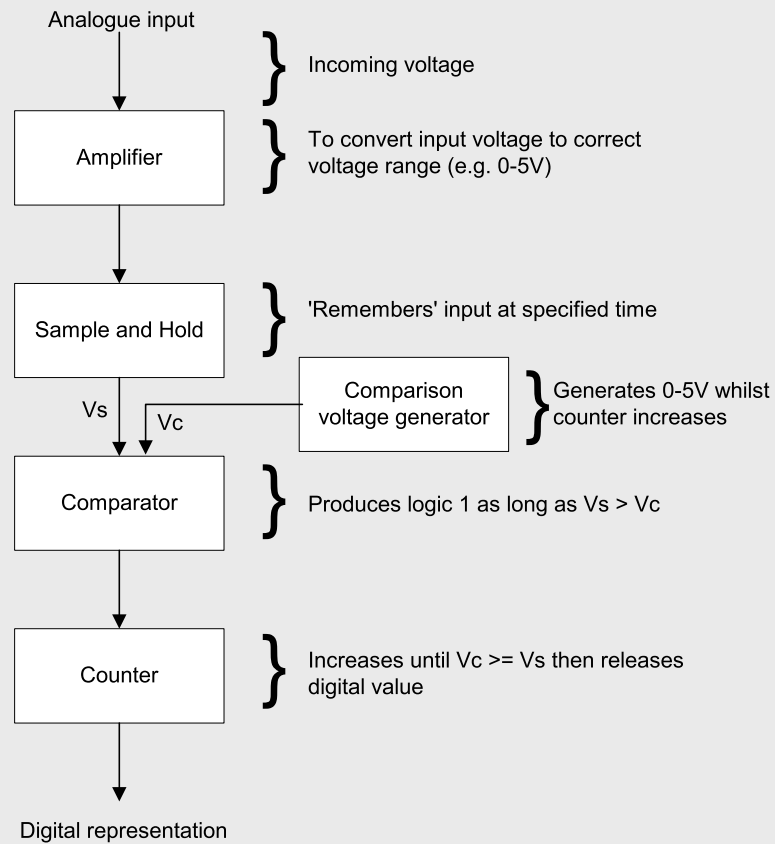
Recording sound

- ▶ First we need to revisit some audio principles
- ▶ Recall that all sounds are made up of one or more sinusoidal components
- ▶ The sound waves are continually varying analogue signals
- ▶ We can record to analogue tape using a microphone
- ▶ But how do we get sounds into a digital computer?

Sampling

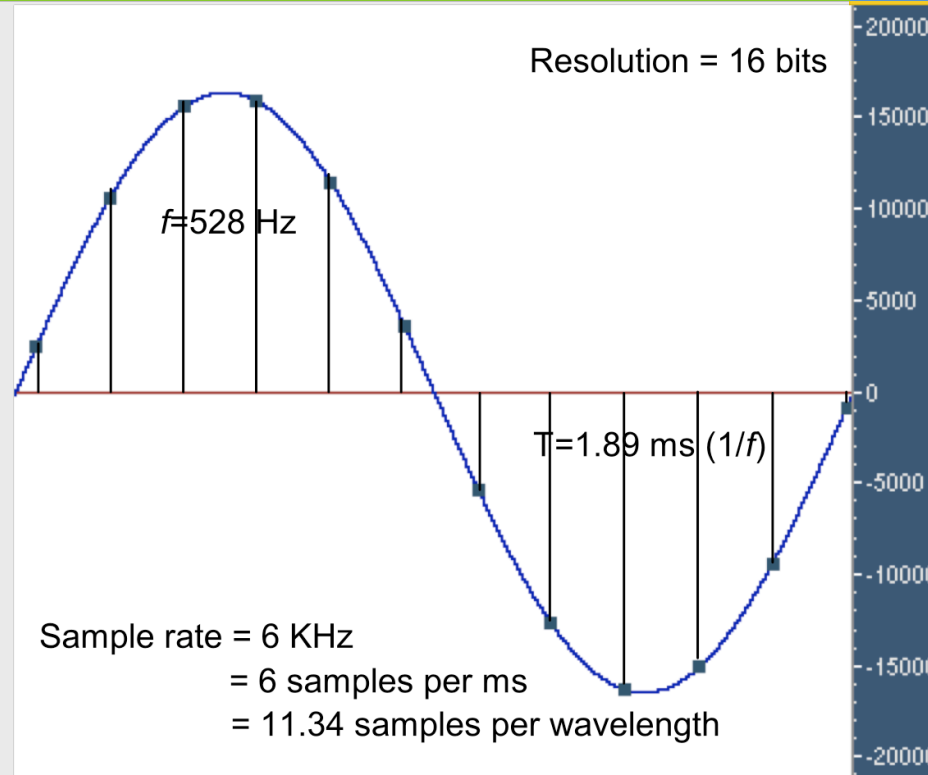
- ▶ The conversion of an analogue audio signal into a digital one is called sampling
- ▶ Analogous to a movie camera running at 30 fps
 - ⊙ Analogue-to-digital Converter (ADC) presented with continually varying analogue signal
 - ⊙ Signal is inspected/sampled at regular intervals and the signal's voltage converted to a binary value
 - ⊙ Sound reproduced by sending binary values (samples) to a DAC

An ADC



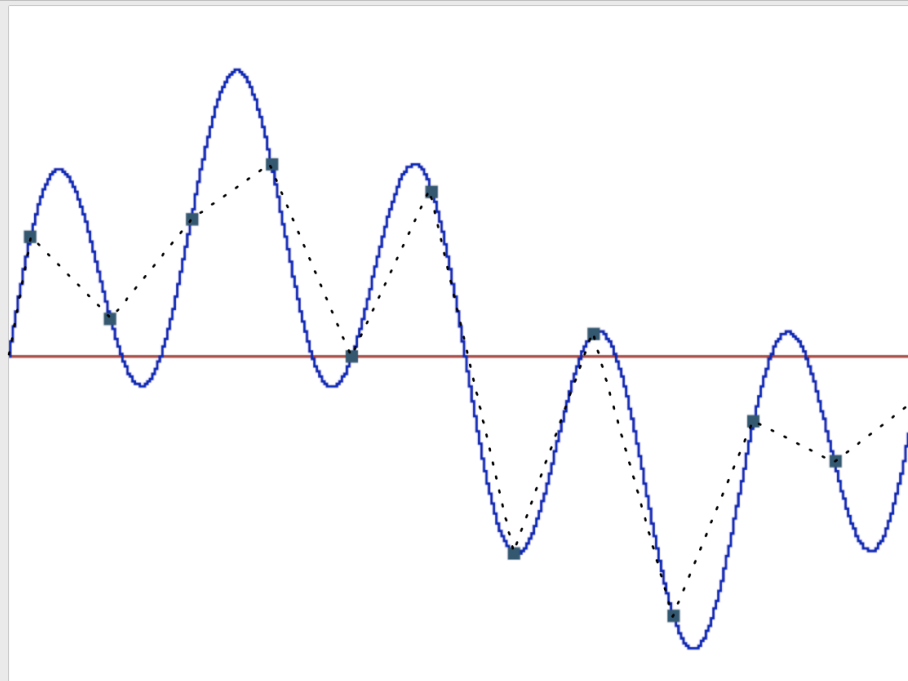
Sampling illustration

- ▶ 528 Hz signal
- ▶ Sampled at 6 KHz
- ▶ Notice the points at which the signal is sampled



Under sampling & aliasing

- ▶ Sampling rate too low
- ▶ Result not accurate
- ▶ Results in aliasing



Aliasing

- ▶ When unwanted artefacts introduced into the signal as a result of too low a sample rate
- ▶ Think of a wagon wheel in a western film
 - ⊙ the spokes appear to move backwards
 - ⊙ Frame rate = 30 fps
 - ⊙ If wheels turning at ≤ 15 rpm, then spokes move forward, otherwise they appear to move slowly backwards

Nyquist's theorem

- ▶ What is the minimum sampling rate that can faithfully reproduce a signal?
- ▶ Nyquist's theorem shows min rate to be twice the highest frequency component of the signal
- ▶ Human ear works in range 20 Hz - 20 KHz, therefore, sample rate should be 40 KHz
- ▶ CD system uses 44.1 KHz to get round problems with filter around 20 KHz

Dynamic range

- ▶ CD system uses 16 bit sampling, meaning the amplitude of input can be divided into 65,536 discrete values
- ▶ What happens if you use lower resolutions (say, 8 bits) to save space?
 - ⦿ 8 bit resolution gives 256 quantisation values
- ▶ Resolution determines dynamic range
 - ⦿ The range of quietest to loudest sounds
 - ⦿ Large the DR, greater your amplitude range, or greater difference between loudest & softest sounds

KB requirements for sampling formats

CM533

Multimedia Time-based Assets

Sampling rate	Mono			Stereo		
	8 bit	16 bit	24 bit	8 bit	16 bit	24 bit
8 KHz	7.81	15.63	23.44	15.63	31.25	46.88
22.05 KHz	21.53	43.07	64.60	43.07	86.13	129.20
44.1 KHz	43.07	86.13	129.20	86.13	172.27	258.40
48 KHz	46.88	93.75	140.63	93.75	187.50	281.25
96 KHz	93.75	187.50	281.25	187.50	375.00	562.50

Combining audio assets

- ▶ When combining audio assets, need to know about their sample rates to ensure proper results
- ▶ Playing back a 22.05 KHz sample at 44.1 KHz will cause it to play twice as fast (and at double the pitch)
 - ⦿ Like playing a 33 rpm record at 45 rpm
 - ⦿ Remember the confusion when 12" singles first came out?
- ▶ You will need to resample

Resampling

- ▶ DOP will only accept audio files in 11.025, 22.05, & 44.1 KHz formats
- ▶ If you have a 16 KHz file you need to resample it
 - ◉ If you import it into DOP it will be converted to 11.025 KHz **without** resampling meaning it will sound slower.
- ▶ Resampling involves re-recording a file as if it were a live source.
- ▶ You can resample at any sample rate you like using Cool Edit Pro

Sequencing

- ▶ The assembling of audio assets along a timeline to create a composition
- ▶ Usually you separate your composition into tracks
 - ⊙ e.g. track for bird sounds, another for drum loops, another for vocals, etc.
- ▶ Sequencers like DOP, Cubase, etc. allow you to mix audio tracks with MIDI tracks
 - ⊙ Though the whole thing eventually needs to be turned into a single audio file

Example

- ▶ “Jingle” in DOP See MIDI & audio tracks

Digital Orchestrator Pro - jingle.ora - [Track View]

Current time: 00:00:00:00 | 001:01:00:00 | 130.00 | 0.00

R	Type	Name	Pch	Inst	Trans	S	M	Reverb	Channel	D
1		Flute sequence	35	TBS Pro Main Synth Out 1	1	00	00	M	<0>	
2		Bass	34	TBS Pro Main Synth Out 1	2	00	00	M	<0>	
3		Synth leads		Microsoft Sound Mapper	25	00			<0>	
4		House piano 1	3	TBS Pro Main Synth Out 1	3	00	00	M	<0>	40 9
5		House Piano 2	1	TBS Pro Main Synth Out 1	4	00	00	M	<0>	50 24
6		Strings		Microsoft Sound Mapper	02				<0>	
7		More Strings		Microsoft Sound Mapper	114				<0>	
10		Face out		Microsoft Sound Mapper	58				<0>	
11		Drums		Microsoft Sound Mapper	05				<0>	
12		Drums		Microsoft Sound Mapper	105				<0>	
15		Drums		Microsoft Sound Mapper	128				<0>	
16		Drums		Microsoft Sound Mapper	124				<0>	
17		Drums		Microsoft Sound Mapper	128				<0>	
18		Drums		Microsoft Sound Mapper	124				<0>	
19		Breaking glass		Microsoft Sound Mapper	07				<0>	
19		Melody		Microsoft Sound Mapper	58				<0>	
21		Flute sequence		Microsoft Sound Mapper	73				<0>	
22		Bass		Microsoft Sound Mapper	50				<0>	
23		House piano 1		Microsoft Sound Mapper	116				<0>	
24		House piano 2		Microsoft Sound Mapper	116				<0>	
26				Microsoft Sound Mapper	05				<0>	

Issues

- ▶ Need to consider pitch and tempo of sources
- ▶ Pitch shifting & time stretching audio is computationally expensive and results aren't always acceptable
- ▶ Big tempo changes in MIDI can sound awful
- ▶ Pitch shifting MIDI data is simple
- ▶ Editing MIDI data simple. Audio is harder to edit

Pitch shifting examples (files on Blackboard)

- ▶ Original file - C Mutes.mp3
- ▶ Pitch shifted
 - ⦿ A Mutes
 - ⦿ D Mutes
 - ⦿ E Mutes
- ▶ Notice that durations remain the same
- ▶ Now listen to C Mutes Octave
 - ⦿ Shift range too big
 - ⦿ Doesn't sound realistic

Time stretching

- ▶ Original file
 - ⦿ Bongo fills 125 bpm (mono).mp3
- ▶ Slowed down version
 - ⦿ Bongo fills 104 bpm (mono).mp3

Other technologies

- ▶ Java Media Framework
- ▶ Beatnik Audio Engine and the eXtensible Music Format
- ▶ Jsyn (Java audio synthesis) at (www.softsynth.com/jsyn/)
- ▶ CSound
- ▶ SuperCollider (www.audiosynth.com)
- ▶ PureData (www.puredata.org)