CHAPTER 35 Practical 12 Bubbles

Aims

Produce the sound of liquid bubbling, taking into account physical factors such as fluid density, viscosity, depth, rate of bubble formation and the nature of the gas source.

Analysis

The bubbles we are interested in here are fluids within other fluids, specifically gas bubbles in a liquid. A bubble is little piece of something that is where it doesn't belong. It doesn't belong there because it's in conflict with its environment and doesn't mix with it. Were this not the case bubbles would either float happily about underwater or the air would redissolve back into the water. On all sides are water molecules pressing inwards trying to crush the bubble. It therefore assumes the minimum surface area possible, which is a sphere. We take water to be an incompressible fluid and the air to be elastic. In this view the air in the bubble is a spring and the surrounding water is a mass. Consider a

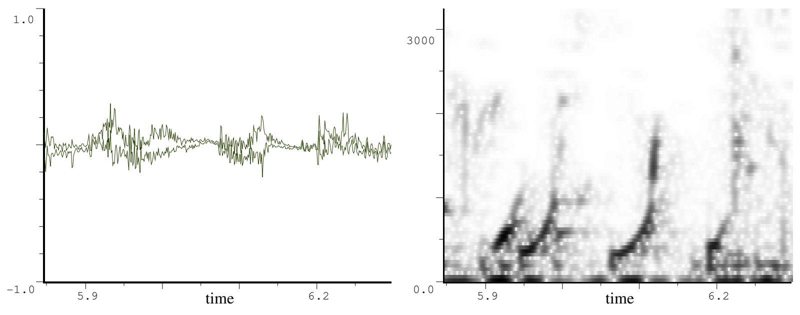

fig 35.1: Bubble analysis

complementary phenomena, a balloon filled with water. If you've made balloon water bombs you understand how they wobble like a jelly. Underwater a bubble wobbles like a jelly too, albeit under a slightly different balance of forces. In Fig. 35.1 we see a the familiar time and spectrogram analysis of a sample sound. It's the sound of bubbles surfacing in a bath tub. The plot was made with a small window for good time resolution, so it looks a bit blurred. Study it briefly now. We will return to this later and it will make much more sense.

Quantisation of regular gas flow

It's quite clear that bubbles are discrete events. Bubbles from some source of gas under pressure appear in a quantised form by a process of relaxation, much like a dripping tap. Recall from our earlier studies of oscillations that a dripping tap and underwater bubbles are kind of complementary phenomena, a case of *quantisation*, where the energy stream is split into packets or quanta (droplets or bubbles here). A dripping tap or bubbling gas under constant pressure release each drop or bubble at regular intervals. One force must overcome another force that resists movement. For a drip the surface tension of the water holds it back. Once the drip becomes big and heavy enough it detaches from the reservoir of water building in the tap and falls under gravity. For bubbles, the driving force is the pressure of the gas and the opposing force is surface tension working to adhere the bubble to the larger body of gas. Pressure around the bubble trying to force it into a sphere will eventually overcome the surface tension and the bubble *pinches off*. However it's rarely the case that water bubbles form under constant pressure, instead they tend to come in bursts which decay in frequency followed by a period of few bubbles, and then another burst. The reason for this involves some complex dynamics, let us just say that once some bubbles have started moving other bubbles find it easier to break through for a short while. We will revisit this concept again when we look at electricity, a phenomenon that shares some things with the behaviour of fluids.

Speed of bubble motion

An alternative way of thinking about the bubble is as a place where there isn't any water, it's not the bubble moving up so much as the water falling down. Wherever the water is it always falls with a constant force of gravity, and so the bubble rises with the same force which we call the *upthrust*. The force exerted on a submerged body by buoyancy equals the weight of displaced fluid, which is Archimedes Principle. Ignoring fluid friction for a moment a constant force causes a constant acceleration (g) which is the gravitational constant, approximately 9.8. So bubbles emerging from the bottom of a pond will get further apart as they rise towards the surface.

Terminal velocity

Additionally, a bubble rising through water experiences forces of fluid friction and turbulence. These create an opposing force proportional to velocity, so just as a body falling in the air reaches a terminal (maximum) velocity, so it is for a bubble when the frictional forces due to its upward velocity match the upthrust.

A rising air bubble in water reaches a velocity of approximately

$$\frac{2}{3}\sqrt{gR} \tag{35.1}$$

for a bubble of radius R. Everything apart from the radius is a constant. What this means to the sound is that whatever the bubble size it quickly reaches its final velocity. Now suppose a bunch of bubbles, some big, some small, were created deep in the water. The bigger ones will arrive at the surface first, followed by the smaller ones.

Size of bubbles

In water the volume of air halves for every 10m of depth, corresponding to a pressure increase of 1 atmosphere. The pressure increases linearly with depth, so the volume of a rising bubble increases as it moves towards the surface. This is not visible in a bath or glass of water. It's an effect that only happens in very deep water, say at least 10m deep. Bubble size can also change depending on how long the bubble remains in a liquid that contains dissolved gases. Where bubbles are formed by cavitation of dissolved gases they tend to grow in size as they absorb more gas, something that happens in fizzy drinks. For practical purposes you can assume that a bubble keeps the same size throughout its life.

Excitation

There are three ways a bubble can be excited to make a noise. When the bubble comes from an underwater source of gas the shock of separation from the larger body imparts an impulse to the bubble. Picture the bubble just the moment before it pinches off by watching the bubbles in a fish tank aeration pipe, it is elongated, but when the bubble pinches it snaps backwards and oscillates. A similar process happens when raindrops or stones hit water, a column of air protrudes momentarily into the water but as the fluid collapses behind it the same pinching occurs. Another kind of impulse is imparted to a bubble during *cavitation*. This is when a bubble simply pops into existence during a pressure or temperature change in a liquid. The mode of this oscillation is slightly different from pinched bubbles since it involves a uniform explosive formation. Finally, there is the singing bubble which obtains its acoustic energy through frictional excitation when rising, these bubbles tend to rise in a spiral or zigzag because of their oscillating exteriors.

Underwater bubbles

The bubble is strongly damped so pinched and cavitated bubbles make only a short sound, less than a tenth of a second. Singing bubbles emit a sine wave mixed with a noisy component due to turbulence. Both of these are very quiet sounds that come from bubbles while they are submerged. When the bubble is much larger, deviations in shape will cause modulation of pitch. Big non-spherical bubbles sometimes sound a bit wobbly while smaller ones sound tightly pitched. Very large bubbles oscillate across two or more axes according to Laplacian equations and exhibit sounds rather like slowly modulated FM. Finally, the perception of pitch depends on the observer. Sounds from underwater don't travel into the air unless the fluid is contained in some kind of tank with thin walls. What we hear in air, where the speed of sound is slower than the liquid in which the event originated, has a different pitch.

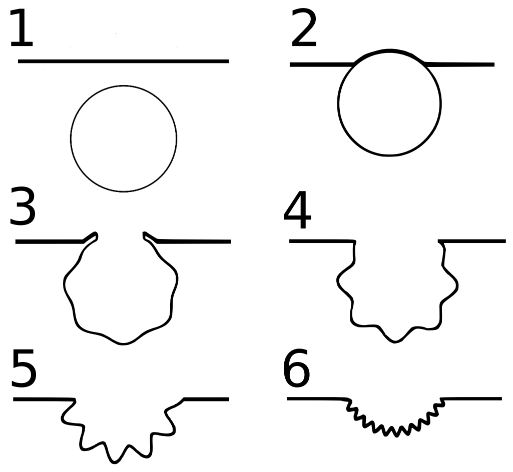

fig 35.2: Surfacing bubble

Frequency

The actual pitch of a bubble depends on a few things. The larger the bubble the lower the sound. But that is a very simple view. The pitch also depends on the ratio of the gas elasticity to the surrounding liquid elasticity, the restoring force, which in turn depends on pressure, which in turn depends on height in the water. The full equations for a bubbles pitch as a function of height, temperature, pressure, fluid density and size are too complex to derive here, even in words, but for the curious the Minnaert Formula

$$f = \frac{1}{2\pi R} \times \sqrt{\frac{3\gamma P}{\rho}} \tag{35.2}$$

relates pressure P, water density ρ, ratio of gas specific heat γ and radius R to frequency. Notice the familiar structure of the equations right hand side, another second order differential system. Experiments give us a value of 3kHz for a 1mm bubble.

Surfacing bubble sounds

What we are actually creating in this patch is the sound of a surfacing bubble, depicted in Fig. 35.2. This is what most people think of as "bubble sounds", not the hissing, ringing or clicks of singing and cavitating bubbles. Excitation comes from the bubble surface being torn apart as the liquid skin maintained by surface tension breaks, forming a Helmholtz resonator.

Because it's a sphere, as it emerges the cavity diminishes, and because the same energy is squashed into an ever smaller space the frequency increases. The result is an exponentially rising sinusoidal wave in the 1kHz to 4kHz range. An idealised graph of this curve (e^x) is shown in Fig. 35.3 for comparison to Fig. 35.1 . Refer back to the spectrogram analysis now and you should be able to identify the exponential rise of the two middle examples. As time moves forward the curve gets rapidly steeper.

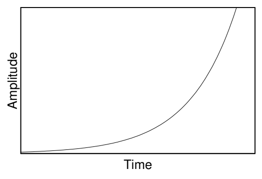

fig 35.3: Exponential rise

Model

An underwater source produces bubbles of varying size which rise to the surface where they pop and ring. This produces sine waves because the oscillation is strongly damped, with exponentially rising frequency because of the changing geometry. The sounds are of a fairly constant duration because the bubble velocities are relatively uniform, thus the time taken to emerge depends only on diameter. The larger (lower frequency) ones emerge first followed by the smaller ones, thus the sound pattern tends to rise in frequency.

Method

The patch is split into two sections so we can decouple the production of underwater bubbles from their sound at the surface. A pseudo-random stream of events is derived from a sequence of small prime numbers. This drives a sound generator based on an exponential envelope and sinusoidal wave generator.

DSP implementation

Bubbling pattern

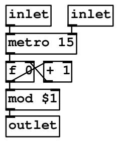

fig 35.4: cycle

The timing part consists of a metronome and counter combined with a modulo operator to provide a circular counter. The abstraction in Fig. 35.4 has two inlets. The first is to start the metronome. The time between each increment of the counter is initially 15ms, which gives a regular timebase, and another inlet is provided to set the period. The cycle range is set by the first abstraction argument, which is substituted in the first argument of mod. We instantiate this object with a value of 200 so it counts between 0 and 199. What we actually want though is not a regular series events. To simulate relaxation of flow a regular random source is inappropriate, which is why we haven't used an approach like that for the fire crackling generator. Something slightly different is called for here. We use a select block to output a bang when an integer between 0 and 199 matches one of its arguments. Do you recognise the numbers in the select block of Fig. 35.5? They are small primes in diverging ascendancy. Humans are very good at picking out patterns, we tend to notice any periodicity in a sequence if we listen to it long enough, but the primes create an illusion of a non-periodic source because they have no common factors.

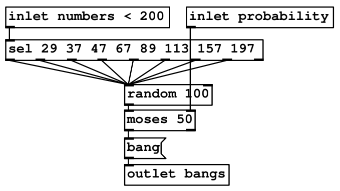

fig 35.5: bubblepattern

Furthermore, having every event produce a bubble would still be too much, so a way of culling a few events is required. Removing one in every two events is sufficient for a realistic bubbling pattern, however we don't just want to remove each alternate event, we want to cull them randomly. By doing this the stream of events will sometimes contain longer gaps and sometimes shorter ones while still retaining the overall feel of a steady average rate. A number between 0 and 100 is generated for each event, and fed to a stream splitter with a midpoint of 50. Because the random numbers are evenly distributed, on average half the events will make it through. Any number that passes through the splitter invokes a bang message. An extra inlet is given to the abstraction to adjust the probability (density) of bubbles.

Sound generation

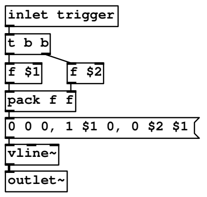

fig 35.6: attack decay envelope

The bubble sound patch will use two envelope generators, one for the pitch and another for the amplitude. First we'll make the amplitude envelope which is a linear attack-decay line. In Fig. 35.6 two floats are obtained from the abstraction arguments, one for the attack time and one for the decay time, both in ms. A bang appearing at the inlet causes both of these to be packed and then substituted in a list for the vline~. It starts at 0.0 then moves to 1.0 over the attack time, and then back to 0.0 over the decay time after a delay equal to the attack time. The result is a triangular envelope peaking at 1.0 and taking a total time of *attack* + *decay*. This abstraction is named `adenv`.

An exponential curve generator is shown in Fig. 35.7. The behaviour of this abstraction, which mimics the geometry of an emerging bubble is at the heart of our sound. A float value for the duration (initially 10 so we don't accidentally produce a loud click) is provided by the first abstraction argument. Upon receiving a bang this number is substituted into a list for vline~ as the time to rise between 0.0 to 1.0. Unlike the linear envelope, we do not use the output of vline~ directly. First it is shaped by the function e^x, made from a constant sig~ and a pow~ object. This abstraction is named `expcurve`.

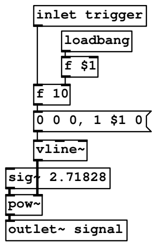

fig 35.7: expcurve

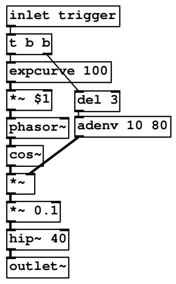

fig 35.8: bubblesound1

Both envelope generators are combined with an oscillator in Fig. 35.8. You can see the exponential curve for pitch is created with a period of 100ms while the linear attack decay envelope has a fast attack (10ms) and slightly slower decay (80ms). So that the amplitude peaks at precisely the right point in the pitch sweep a delay is added before it triggers. To have bubbles at different pitches we scale the pitch envelope by the first abstraction argument, typically this will be between 1kHz and 3kHz. The final output is scaled by 0.1 to get a sensible volume and a high-pass filter removes any very low frequencies.

Putting it together

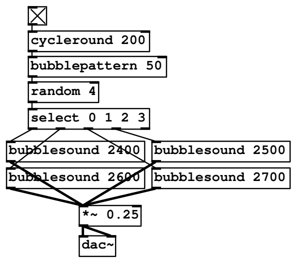

fig 35.9: Several bubbles

This part is easy, all we have to do is connect the event generator to the bubble sound generator to get some results. If several instances of the bubble sound with slightly different frequencies are connected via a random selector we get a more interesting effect. We haven't dealt with the issue of bubble size yet, so we assume fairly consistent bubble sizes with similar frequencies. If you tweak the parameters in the above example you will notice that attack and pitch are linked, moving the attack also alters the apparent pitch of the bubble. This codependency is a feature of the simple model we've used where moving the attack changes the point at which the amplitude peaks during the pitch rise.

Polyphony

Listening to the patch above you will quickly notice a problem. By randomly selecting which bubble will sound from uniformly distributed numbers there's nothing to stop the same bubble generator being picked twice. If this happens before it has finished playing then re-triggering the envelopes causes the sound to be abruptly truncated. Another limitation is that we only have four bubble pitches entered as fixed abstraction parameters. It would be nice to have a source of bubbles that are not only random in time without the possibility of being cut short, but random in pitch too.

Bubble factory

The next patch rewrites the bubble sound and pattern generator to give control over the density and average pitch. It uses *round-robin* allocation. This allocates in repeating order, 1, 2, 3, 4, 1, 2, 3, 4... With this method we can be sure that no bubble will cut short the previous one so long its duration is less than the time to cycle round all generators. Two more improvements can be made. When the bubble pitch is high (for smaller bubbles) the duration should be proportionally shorter, and since it contains less energy (volume) it should be quieter. From a common parameter, bubble size, we will calculate the pitch, duration and amplitude as separate functions.

Here I've redesigned the envelope to show an efficiency. The curve obtained using pow~ can be approximated for short sounds using only multiplies, which is a little more efficient. A line generator rises immediately to 1.0 and then falls to zero in a time given by a value passed through the inlet. The complement of a quartic curve appears on the first outlet, which will be used as the bubble pitch envelope. By taking the square one more time we obtain the 6th power, which falls to zero a little quicker than the pitch rises. This envelope replaces the exponential rise and attack-decay used previously. Results aren't quite as nice as before, but it shows how a patch can be simplified where a less accurate model will do. This dual output envelope is subpatched as `pd env4pow`.

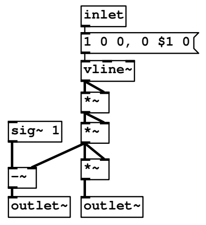

fig 35.10: Two curve envelope

Each bubble sound is going to be created by a number ranging from 0.0 to 1.0 that represents its size. So, next we redesign the bubble sound to use the new envelope generator, with amplitude, base frequency and duration dependent on the supplied size parameter. A float value appearing at the inlet is distributed to three functions. The first (left branch) multiplies the size factor by 90ms to get longer sound durations for bigger bubbles. In the centre branch we obtain the pitch value, which is the complement of bubble size (bigger bubbles having lower pitch). A scaling factor of 300Hz and an offset of 100Hz are applied to a fixed base of 2kHz. Finally we obtain an amplitude factor proportional to the bubble size (right branch) and add a little offset so that even small bubbles make a sound.

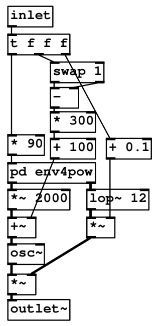

fig 35.11: Bubble with parameters

Finally we combine four of these generators into a polyphonic bubbling sound. We need a metronome to create events. As an aside, in my opinion this method is lacking. It would be nice to improve it here, but we must move on. We've already looked at a time distribution based on prime numbers, but you may like to refer forwards to the Poisson time distribution examined in the chapter on raindrops. To obtain randomly distributed bubbles a uniformly distributed number between 0 and 100 is compared to an `intensity` inlet (initially 70) to see whether to generate a bang. Each bang advances a cyclic counter, the output of which becomes the first element in a list used to route the bubbles to generators. At the same time we generate a small size variation which is added to the `size` inlet. Two element lists passed to route are stripped of their first element. The remaining float is sent to one of four possible outlets, each of which activates a bubble generator. The size inlet should be between 0.0 and 1.0, and the intensity inlet should be between 70 and 99. This patch will be used in a later example for the bubbles made by poured liquids.

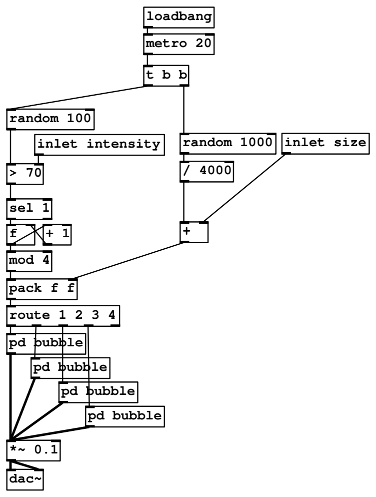

fig 35.12: Polyphonic bubble factory

Results

Source . . . . **http://aspress.co.uk/sd/bubbles.html**

Conclusions

We've taken a good look at the physics of bubbles and sounds produced by the changing shape of an air cavity. Pitch is affected by the bubble size, and the rate of pitch change by the speed of the surfacing bubble. The behaviour of bubbles is determined by several factors like the density of both fluids. Bubbles from a steady source of gas behave like a relaxation oscillator.

We've seen how prime numbers can be used to produce intervals that don't sound periodic, thought about the effects of random number distributions, seen how to use the pow~ object to create an exponential power curve, and how a more efficient approximation can sound almost as good.

Additionally we have seen how to build deeper abstraction stacks where the arguments of higher ones are passed in as the arguments of deeper ones. And we have looked at polyphony, where more than one instance of the same sound happens at the same time. The problem of voice management has been explored and we saw why round robin voice allocation is a better idea than triggering instances randomly because the same instance may be triggered again before it has finished.

The experiments and formulas of Newton, Reynolds, Davies, Taylor and

Stokes are all important to understanding bubbles, so some are given in the references section. You are encouraged to read these, and the work of Van den Doel to understand a more elaborate model before moving on to the next practical on flowing water.

Exercises

Exercise 1

From the Minnaert formula, how do viscosity and temperature change the sound of bubbles? What would you expect to hear from bubbling lava and boiling water?

Exercise 2

Wrap the bubble factory in a control structure so that bursts of bubbles emerge according to their size.

References

Ucke, C., Schlichting, H.J (1997) "Why does champagne bubble?", Physics and Technology Quest Journal, Vol. 2 pp. 105-108
Walker, J. (1981) "Bubbles in a bottle of beer, Reflections on the rising", Scientific American, Vol 245, pp. 124
Stokes, G.G. (1851) "Turbulence of a sphere in a Newtonian fluid"
Leighton, T. G., Walton, A. J. (1987) "An experimental study of the sound emitted from gas bubbles in a liquid" Eur. J. Phys, Vol. 8, pp. 98104
Leighton, T. G. (1994) "The Acoustic Bubble" Academic Press London

Acknowledgements

Thanks to Coll Anderson for the bubbles sample used during analysis.