## Getting the sounds to alternate

The ternary operator is an operator and therefore should give a result. We have to do something with this result. In general we assign this result to a variable: we associate it with a name we can use later. Let's call this variable freq, short for frequency. This frequency will go into the equation later on. We should assign freq1_in_Hz in some cases and freq2_in_Hz in the others. So we have the following for our code:

```
freq = freq1_in_Hz if [condition] else freq2_in_Hz
```

The question now is the condition. We know that as time goes by it should change. Hence the condition depends on time. We also know it should alternate between two values. From class we know the if time were an integer, `time % 2` would alternate between two values: 1 and 0. One option becomes then `int(time) % 2`. This indeed alternates between 1 and 0. However, it takes a full second to alternate. If time went 4 times faster though, we would get what we need. Let's try therefore multiplying time by 4: `int(4 * time) % 2 == 0`. Make sure you see why. This is the right condition. Test it in your code.

A different approach is based on figuring out how many intervals of 0.25 s have gone by. We do this by using the code `time // 0.25`. This will count the intervals as time goes by: 0, 1, 2, 3, 4, 5, 6, 7, etc. Try it. To make something that alternates from this we use `(time // 0.25) % 2` as we studied in class. So the final code is `(time // 0.25) % 2 == 0`. The parenthesis are unnecessary due to precedence.

## Getting the equations for the factors in Task 3.2

Notice that the total duration is divided into three equal periods. Hence each period lasts duration/3. One way to obtain the equations is to observe that they are straight lines. They also depend on time. You can stop here and try to derive them based on what you have learned about straight lines or linear equations and how to get them with two points. Use the start and the end of the period. Keep reading for a simplified process. Skip to the end to see the actual equations.

## Simplified process

**Left channel**

During the first period, the left channel, which ends at duration/3, has a factor that increases as time increases. (See the figure in the updated assignment.) Hence it is proportional to time: our first attempt can be `left_factor = time_elapsed`. Let's test it a bit.

When the time reaches the end of this period the factor has to be 1. If we leave our first-attempt equation, it will yield left_factor = 4 in the example given (duration = 12 s → end of first period at 4 s). It should be 1 instead. Hence we need to divide by 4. In general we need to divide by duration/3. (Test the beginning and the end.) Now it works. This is the left_factor we should use for the first period.

(Try the second period if you want. The details are below if you rather observe.)

For the second period the slope is negative. The factor should decrease as time goes by. As a matter of fact, it should decrease with the same slope it had when increasing: compare how long it takes it to go from 0 to 1 to how long it takes to go from 1 to 0. We get the following as a tentative equation `left_factor = - 3 * time_ elapsed / duration`. However for the example in the assignment at time_elapsed equal to 4, it should give 1. It gives -1 instead. Wrong but on the right track: it has negative slope. If we add 2 to get the 1 we need as a result, the equation becomes `left_factor = 2 - time_elapsed / duration`. This evaluates to 1 when time_elapsed is 4 and to 0 when time_elapsed is 8, as expected. We got the second period covered. The third period is easy `left_factor = 0`.

(Try the right factor on your own if you want. Continue reading for a description.)

**Right channel**

For the right channel the first period is easy: `right_factor = 0`. The second period is more challenging but we can start with a similar approach to the one used before. The factor increases in time as before and it seems to have the same slope.

Our initial attempt is `right_factor = 3 * time_elapsed / duration`. At 4 s it gives 1. It should have given 0 though. We can subtract 1 to see if that solves the issue. The equation becomes `right_factor = 3 * time_elapsed / duration - 1`. Success! This works for 4 s and 8 s. Check it yourself.

Using the same approach as we did before, we can assert that the slope is negative but of the same magnitude as before. So `right_factor = - 3 * time_elapsed / duration`. For time 8 s this yields –2. It should have been 1. Adding 3 to fix this we get our final equation `right_factor = 3 - 3 * time_elapsed / duration`. See below for a graphical representation.

## Graphical representation

The equations for the three periods are:

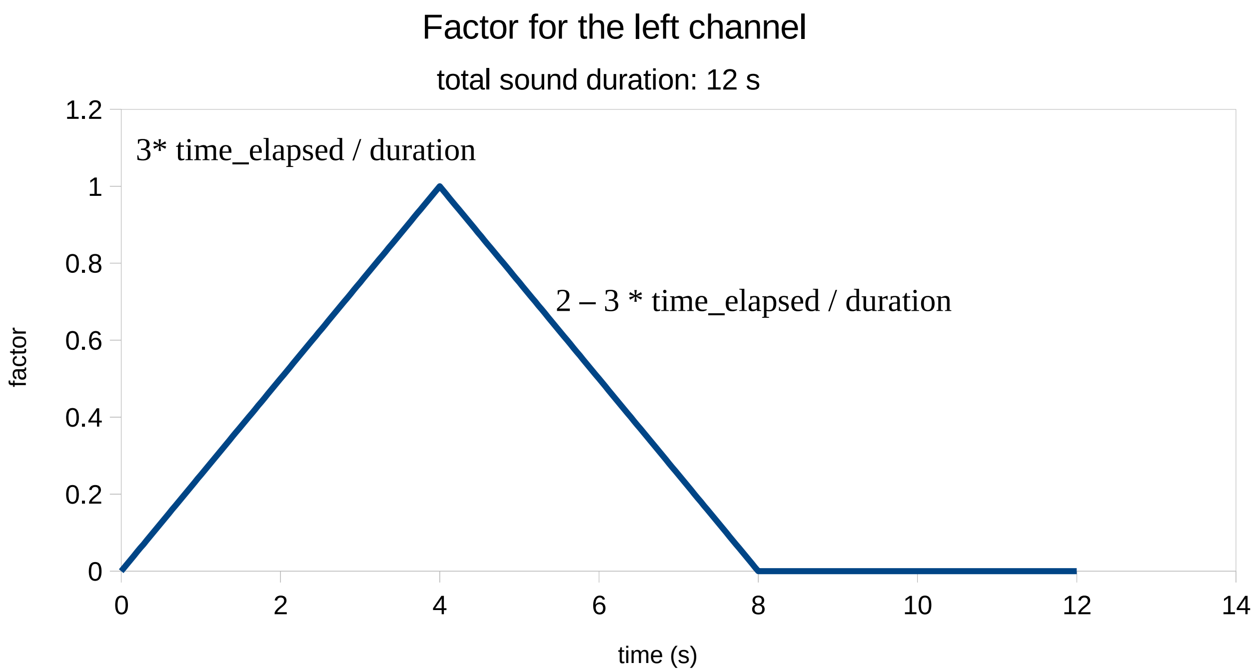


*Figure 1. Multiplicative factor for the left channel as time progresses.*

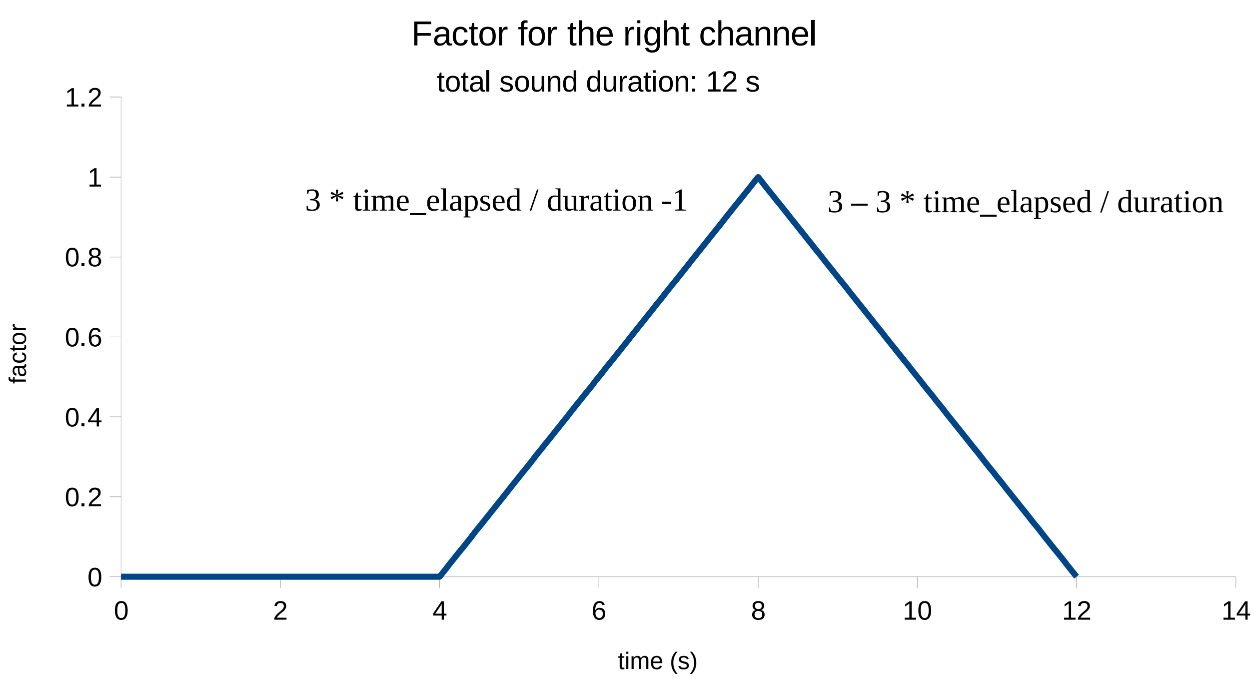


*Figure 2. Multiplicative factor for the right channel as time progresses.*