



The Wersi Sound Factory Master Classes

Master Class II : Winter 2016 : The Matrix - Part 5 : Source/Modifier Components - C

Jeff and Ian have provided a more technical explanation of some terms in an Appendix at the end of this Master Class

The Source/Modifier List for MC11

LFO 1
LFO 2
LFO 3

Meaning of 'LFO'

The unit 'Hertz' [Hz] used to be called 'cycles per second' [c/s], a more explanatory description than 'Hertz'.

Differences between LFO 1, 2 and 3

A more technical explanation of 'The Low Frequency Oscillator' (2.) can be found in the Appendix at the end of this Master Class

In this MasterClass we shall be continuing down the list of Source and Modifier Components as given in the Sound Factory Manual. The next three items on that list are: Scale 2, Arp 1 and Arp 2. At this time, these particular choices of modulation appear to be unassigned in function. We shall, therefore, move on to the next parameters in the list: the group of LFOs.

LFO stands for 'Low Frequency Oscillator'. In case you didn't know, the human ear can detect only a limited range of frequency values compared to what is actually possible to produce. In general it is considered that humans have the ability to hear frequencies from 20 Hertz to 20,000 Hertz, the audio bandwidth of a CD. The range itself will vary from person to person because of factors that include age, size of head, ear canal length as well as pinna size/shape/angle, density of ear drum and so on.

Low Frequencies are waveforms of less than 20 Hertz: these are considered to be 'Subsonic'. We can't normally hear them but we can hear their effect on other waveforms that are in the range of hearing as well as feel their presence when output via a subwoofer speaker, for example. An LFO is an oscillator that can produce either a cyclic waveform or a random set of values at a chosen rate. What we are interested in is the effects that they can produce when used to modulate the various destinations available in the modulation matrix.

There are three LFOs available and they are Electronic and Bi-Polar. Before we get onto some experiments to try, there are some differences between the LFOs themselves that need to be understood.

First of all, LFO 1, and LFO 2 are 'Global' within a sound. What that means is that each layer that uses either LFO 1 or 2 will be synchronised to each and every other layer that uses LFO 1 or 2. There will be an experiment to show this in the next MasterClass 12.

LFO 3 is different from the first two in that each individual sample layer in a sound can contain its own individual/dedicated LFO 3 parameter settings, with no interaction between layers. Another important point is that LFO 3 also appears twice as a possible destination source in the Matrix itself. This gives LFO 3 a lot more capabilities than LFO 1 and 2 as far as flexibility is concerned. Once again, there will be some experiments to try to help understand just some of the possibilities.

First we are going to identify the actual effect on a sample of choice. So load into 'Selector 1' the following: Violin 1 * (090-000-008). Make sure that Aftertouch has been activated for this selector in your Selectors/Advanced options and that Selector 1 is the only Selector sounding.

This has been chosen because it has just a single sound layer to edit and also because it presents us with our first experience of using 'Biasing', also known as 'Side-Chaining', modulation techniques.

'Side-Chaining'
explained

See also
MasterClass 8,
page 3

Side-Chaining is the use of either a physical and/or an electronic source to control another physical or electronic source. For example: a light switch is a controller and has only two states, on or off. A dimmer switch between the activation switch and the light bulb is an example of side-chaining. Not only can the light be on or off but it can be set to be anywhere in-between those two possible values: that's how side-chaining works.

In Sound Factory, the side-chain function is called a 'Modifier' and its possible value range is called 'Modify'.

Now play any key and keep it held down. You should notice a periodic shift in the frequency appear just before half a second has gone by. This is called 'Vibrato' (also known as FM - Frequency Modulation) and is produced by using a LFO to modulate the pitch of the sample.

Matrix for
Violin 1
explained

Now go into the Sound Factory/Expert Edit/LFO-Matrix page. The modulation matrix for this sound may seem a bit complicated at first but if you have been following the other Master Classes you will know that we can break this down into smaller, more manageable parts to understand and process. There are three lines of values, so let's look at and deal with each row in turn.

Row One (Original).

	Source	Modifier	Modify	Destination	Depth
1	LFO 3	Delay Ramp	+462 ms	Pitch	+12.0%

First of all, LFO 3 is the chosen modulation source. This is handy because all the available parameters for controlling/setting this option are contained towards the bottom of the matrix page in four boxes in the 'Layer LFO3' panel. (Remember that this LFO is able to be sent only to destination parameters that are contained on this layer's matrix page. In other words, it is 'Local' and not 'Global' in function).

Delay Ramp
explained

The 'Delay Ramp' is our first example of side-chaining a modulation with another modulation. Think of this as the output of the LFO is being controlled by the Delay Ramp (Modifier). The value in the box to the right (Modify) controls time and also has a negative or positive sign. Negative time values cause the delay ramp to produce an inverted output control function. A description of this modulation will be repeated in a later MasterClass, but to help understand at this moment, here's how it works.

More technical
explanations of
'Delay Ramp' ^(1.)
and
'Modulation' ^(3.)
can be found in
the Appendix at
the end of this
Master Class

The Delay Ramp is basically a straight line with two points: the 'Start' and 'End' points which are thought of as either an increase in value from zero (start point) to maximum (end point), represented here as '0' to '1' or, zero (start point) to a negative maximum (end point), represented as '0' to '-1'. The time it takes to get from point to point is controlled by the value box (on the right of it called 'Modify') and has a control range which starts in 'ms' (milliseconds) and converts to seconds at larger values (32 seconds being the largest amount of time available to get from zero to maximum).

How to
temporarily
disable Matrix
Rows

To demonstrate this, temporarily disable rows two and three. You can do this either by selecting 'Off' in rows two and three 'Source' or 'Destination' boxes. In this example the 'Destination' box method is the quicker of the two choices as the Destination list at present is relatively short by comparison to the others. So rows two and three should now have the word 'Off' in their Destinations.

Play any key now and the only active modulation is the first row's values - the vibrato is taking a small amount of time to 'ramp up' to its maximum value from a zero vibrato start point. We can change those values assigned in the parameter boxes and listen for the outcome.

First change the Delay Ramp time value from +462 ms to zero (+0 ms). By playing the keyboard we find that the vibrato (LFO 3) is active from the very first touch of the note sounding. In theory, the Delay Ramp went from the start point (zero) to the end point (maximum) in zero ms (+0ms), or in no time at all.

Next, increase the values in the Modify parameter box to read '1000 ms' or, in other words, one second. Play any key, hold it for at least one second and the vibrato will be increased from zero (its start point) to its maximum (the end point) over a one-second time period. Try even larger values for even longer times. Remember to hold the key down for at least as long as the time you have specified or you will not hear the modulations take effect.

The other thing to try here is setting a negative time value, for example '-1.00 s'. Because the delay ramp has had its output inverted, the control works in reverse. In this case, the LFO 3 is present from the keydown and then its effect is reduced by the delay ramp over a period of a second until no modulation from LFO 3 can be detected. The vibrato therefore gradually vanishes during that time period.

Another option is to combine two rows with differing settings but have both rows going to the same destination. For example, in row one insert these values:

Row One (Modification 1)

	Source	Modifier	Modify	Destination	Depth
1	LFO 3	Delay Ramp	-1.47 s	Pitch	+12.0%

Then, in row six add this data:

Row Six (Modification 1)

	Source	Modifier	Modify	Destination	Depth
6	LFO 3	Delay Ramp	+4.01 s	Pitch	+12.0%

Now there's only one value change in these two rows, but if you play any key and hold it for long enough (at least 4.01 seconds) you will hear that the vibrato starts from the keydown and then fades out during the first 1.47 seconds leaving the Violin sample without any vibrato for a small period of time until LFO 3 is added back fully into modulating the pitch again from plus 4 seconds onwards. You've created a basic envelope control using two lines of code and the delay ramp option as a side-chain controller. More on this in future MasterClasses: let's get back to working with LFOs.

Now set rows 1 and 6 Destinations to 'Off' and let's examine the original values in row 2. Return row 2's Destination to 'Pitch', thus:

Row Two (Original)

	Source	Modifier	Modify	Destination	Depth
2	LFO 3	Mod Wheel	+98%	Pitch	+12.0%

Basically this uses LFO 3 being side-chained by the Mod Wheel. Play any key and move the Mod Wheel forward to hear the effect. This allows the player to add more modulation via the MW at any point he/she feels that it is required. Return row 1's Destination to 'Pitch' and play any key, adjusting the Mod Wheel also. Notice by using the MW together with the original first row of values adds to the depth of the overall modulation amount.

So far we have established that we can control the time taken for a modulation (in this case, LFO 3) to be added or subtracted to a destination via the Delay Ramp option, but what about the frequency of the modulation (speed of the vibrato) itself?

With LFO 3 we have two options to change the modulation frequency (vibrato speed).

1. In the matrix destination list. LFO 3 has two entries in this list; 'LFO3 Rate' and 'LFO3 Depth'.
2. From LFO 3's dedicated 'Rate Hz' parameter box found in the 'Layer LFO3' table at the bottom of this LFO/Matrix page. Because of these two differing

Two options for changing the LFO 3 modulation frequency (vibrato speed)

access points into the 'LFO 3 Rate' they interact with each other and need to be set carefully to get the required results. Look at row three's values:

Row 3 (Original)

	Source	Modifier	Modify	Destination	Depth
3	Aftertouch	Off	-1.0%	LFO3 Rate	+85%

A quick inspection of these values reveals that Aftertouch is routed into LFO 3's 'Rate' control. There is no 'Modifier' (side-chaining) present and therefore the 'Modify' value itself (-1.0%) is unimportant and not used. However, the remainder of the values are all valid and show us one way we can change the frequency rate of LFO 3 in real time.

What I'm pointing out here is that one value will affect the other and vice versa. The 'Rate Hz' parameter has a basic speed setting (4.47 Hz) and then, in the Matrix, a modified speed setting. The 'Rate Hz' parameter below the matrix should always be set first as this is the one that governs the fundamental frequency of the LFO. Any modifications that need to be done in real time, however, should be coded into the matrix for real time control. Do not change the 'Rate Hz' parameter in LFO 3 in this experiment. You cannot set a value of zero for this parameter; it will always either be representing Hertz/frequency or time taken to complete a cycle in ms/seconds.

Rows 1, 2 and 3 should now be active (not row 6) in their original form. Press any key lightly down so the note will sound and then listen for the LFO to be introduced via the delay ramp code line. Then after the modulation has started, press further into the key (Aftertouch). Notice that the rate of modulation changes. If you want to make this more obvious to your ears then change the code to this:

Row Three (Modification 1)

	Source	Modifier	Modify	Destination	Depth
3	Aftertouch	Off	-1.0%	LFO3 Rate	+100%

Now repeat the previous experiment - press any key lightly and then increase the Aftertouch pressure. The increase in vibrato speed owing to the Aftertouch should be more obvious.

You could of course change Aftertouch to Mod Wheel for example:

Row Three (Modification 2)

	Source	Modifier	Modify	Destination	Depth
3	Mod Wheel	Off	-1.0%	LFO3 Rate	+100%

Move the Mod Wheel forward and the LFO 3 frequency (vibrato speed) increases.

If you wanted the Mod Wheel to slow down the LFO 3 frequency, change the values to read as follows:

Row Three (Modification 3)

	Source	Modifier	Modify	Destination	Depth
3	Mod Wheel	Off	-1.0%	LFO3 Rate	-100%

The remaining LFO 3 matrix option (LFO3 Depth) can be used to add to or subtract from the LFO 3 modulation Depth range in real time. The following row values set up the experiment to try:

Row Three (Modification 4)

	Source	Modifier	Modify	Destination	Depth
3	Aftertouch	Off	+0.00%	LFO3 Depth	+100.0%

This, with the combination of the original rows one and two values, will increase the depth of modulation when Aftertouch pressure is applied to the keyboard. The positive sign values add to the depth range.

However it appears that the negative Depth value has more impact than the positive version, so after trying Row Three (Modification 4) change row three to read this:

Row Three (Modification 5)

	Source	Modifier	Modify	Destination	Depth
3	Aftertouch	Off	+0.00%	LFO3 Depth	-100.0%

Now the LFO3 modulation disappears when we apply Aftertouch pressure to the keyboard. The negative sign values subtract from the modulation effect.

Now with this being Sound Factory we could also have both a 'Rate' and a 'Depth' increase/decrease in real time if you required it. You just need to add these values to the row four as well:-

Row Four (Modification 1)

	Source	Modifier	Modify	Destination	Depth
4	Aftertouch	Off	+0.00%	LFO3 Rate	+100.0%

Depending on your sign values for Depth in rows three and four, you will get different results. Set the sign values to what is required and when finished programming save it as a new sound (eg Violin Vibrato).

An extra point here is, in general, it would appear that it is a really good idea to program LFO 3's rate from its dedicated parameter control (the 'Rate Hz' box in the 'Layer LFO3' table) and use the Matrix only to modify the depth/rate in real time. The next MasterClass 12 will deal with those dedicated parameters and the use of LFOs in the Matrix for going beyond vibrato functions.

Jeff's Delay Ramp/Pitch Depth Demo

Jeff has provided this demonstration of how the Delay Ramp and Pitch Depth parameters can be used to modify an instrumental Whistle to that of the human voice.

There are a number of instrumental Whistle sounds in the Wersi Sound Database but none that are available for the human voice. We will use the Sound Factory Matrix to modify an instrumental Whistle sound to that of a Vocal Whistle. Here's how it's done.

Creation of a Vocal Whistle Sound

- 1) Load the Whistle * sound (091-000-035) into Sound Factory. There is just one sample, namely Whistle.
- 2) Select the Matrix page for this sample and locate Row One:-

	Source	Modifier	Modify	Destination	Depth
1	LFO 1	Delay Ramp	250 ms	Pitch	+12.0%

The Vocal Whistle requires a continuous vibrato with a larger pitch variation, so modify the line to read:-

	Source	Modifier	Modify	Destination	Depth
1	LFO 1	Delay Ramp	0 ms	Pitch	+15.0%

- 3) To make the Vocal Whistle more realistic we can add a second Whistle sample and apply a degree of detuning. Return to the Sample page and copy the existing Whistle sample. Now for the first sample set its Cent value to +15.7 and for the second sample set its Cent value to -15.7.

On the WersiClub International website Sound Factory page there is an audio demonstration of this sound featuring the 'Skye Boat Song'. Just a word of warning though, the recording contains nautical effects from the Wersi Special Effects Unit, so if you are prone to sea sickness I recommend taking a pill before listening!!

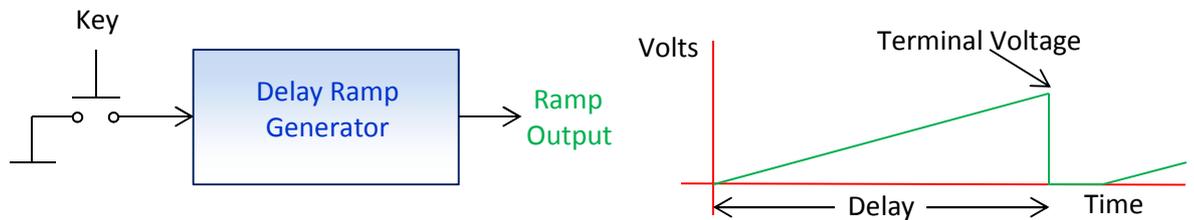
Ian Terry
Jeff Ormerod
Colin Moore

January 2016

Appendix

1: [The Delay Ramp](#)

The term Delay Ramp originates from the days of analogue synthesisers and relates to an electronic circuit that was designed to implement a delay prior to a required function being activated. In our Sound Factory examples we are using this parameter to generate a delay between a key being pressed and a Low Frequency Oscillator (LFO3) being activated. We can depict this diagrammatically as shown below.

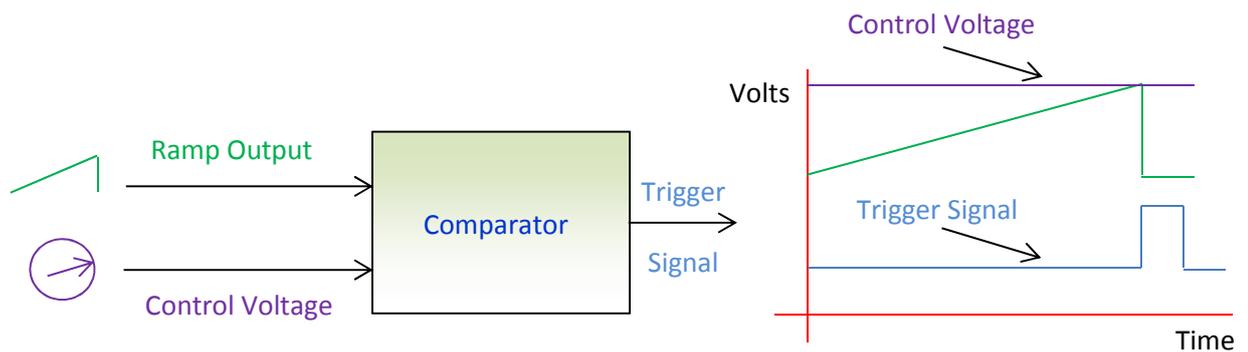


Analogue Implementation

When a key is pressed the Delay Ramp Generator produces a ramp waveform at its output that continually increases from 0 volts until a Terminal Voltage at the top of the ramp is reached. The time taken for the waveform to reach this voltage is our required delay, and this corresponds to the value that we enter into the Delay Ramp 'Modify' box of a Sound Factory Matrix row. For the Delay Ramp to be of practical use in an application such as our pitch modification example it should be able to provide the following :-

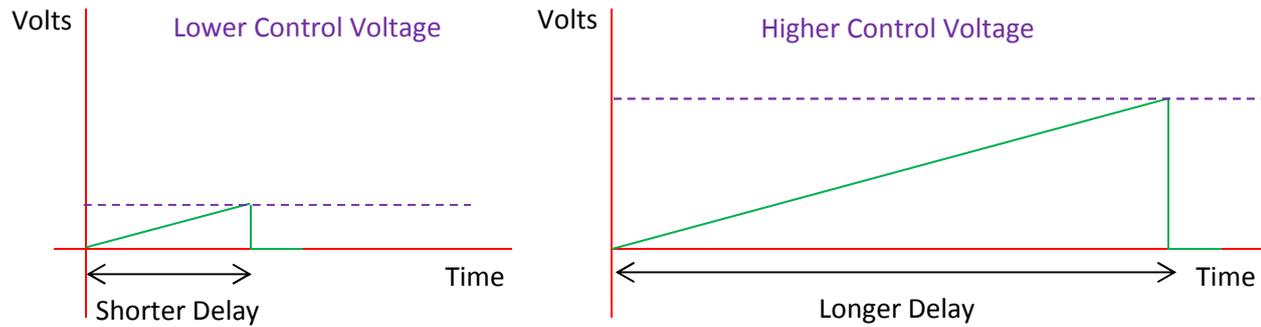
- A Trigger Signal at the Terminal Voltage to active our required function.
- A reset to 0 volts to repeat the cycle when a new key is pressed.
- An input facility for a variable delay time.

One way of accomplishing these requirements is to connect the output of the Delay Ramp Generator to an electronic component called a Comparator. This continually compares the varying ramp voltage to a constant voltage we can call a Control Voltage representing our required delay. This is shown diagrammatically below.



These two voltages will be equal when we reach the Terminal Voltage. At this point the Comparator will generate a Trigger Signal which can then activate the required function. So all we need now is a means of specifying the Control Voltage, and since we require different delays, this needs to be variable. A variable resistor known as a Potentiometer can perform this function and could be installed on the synthesiser console as a user adjustable delay control. The higher the Control Voltage the longer it will take for the Delay Ramp to reach the Terminal Voltage and therefore the longer the delay will be. Conversely the lower the Control Voltage the quicker the Terminal Voltage will be reached and the shorter the delay will be. The Trigger Signal can also be used to reset the Delay Ramp Generator.

The following diagram shows the effect of varying the Control Voltage.



Software Implementation

Digital software implementations of the Delay Ramp like the Sound Factory operate in a completely different way, but they exhibit all the same characteristics:-

- A facility for defining a variable delay
- A facility for implementing that delay
- A facility for generating a Trigger Signal

A common technique in software for implementing a delay is to continually decrement a digital count until it becomes zero. The size of the count will determine the delay, and the real time taken to decrement this down to zero will be the actual delay. The following conceptual computer code illustrates this technique.

Program Code	Comments
LOAD COUNT	Load the delay count
LOOP DECREMENT COUNT	Subtract 1 from the delay count
IF COUNT = ZERO THEN END ELSE LOOP	If the count is zero jump to END If the count is not equal to zero then jump back to LOOP
END OUTPUT TRIGGER SIGNAL	Generate the trigger signal

The value specified for the delay together with the speed of the computer's clock and the real time taken to execute each of the instructions in the code will all be used to calculate a suitable value for the delay count to achieve the required delay.

So we can see that modern software implementations of the Delay Ramp are far removed from the original analogue realisation, but the terminology continues to be used even though the technology has substantially changed.

2. [The Low Frequency Oscillator](#)

A Low Frequency Oscillator (LFO) is an electronic circuit that produces a regularly repeating waveform typically at a frequency lower than 20Hz. It can be used in analogue synthesisers to modify or Modulate another waveform. In our Sound Factory example we are using this feature to vary the pitch of a sound sample (Violin) by modulating its frequency.

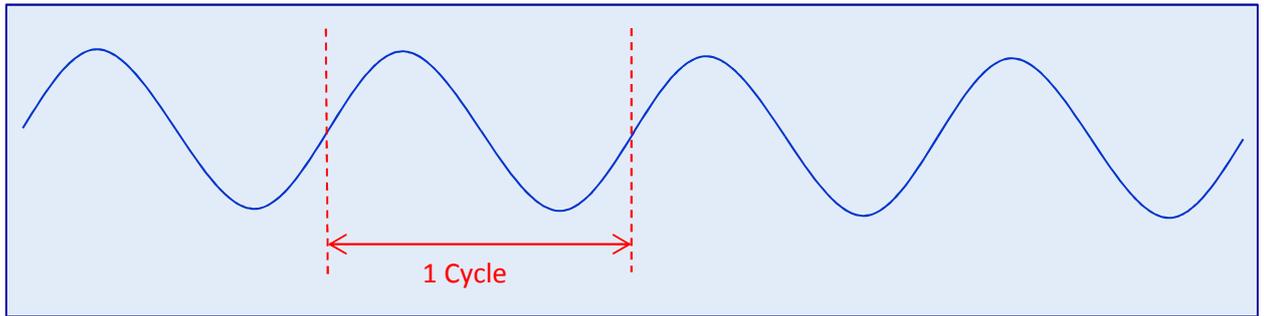
Analogue Implementation

An analogue LFO oscillator circuit is designed to generate a particular type of waveform. By convention we depict a waveform as a graph of voltage on the y axis against time on the x axis. The Sound Factory provides three LFOs, namely LFO1, LFO2 and LFO3. Each LFO can provide Sine, Triangle, Sawtooth and Square waveforms across a frequency range of 1.07Hz to 30Hz, and a time range of 1 sec to 30 secs which enables an LFO to be synchronised to

the tempo of the rhythm unit. There are also two additional LFOs, Sine Amp and Sine Freq, available as 'Modifier' parameters in the Matrix. The LFO waveforms are depicted below.

The Sine Wave

This is a smooth continuously varying waveform.



One repetition of the waveform is known as a Cycle. The number of cycles per second is the Frequency of the waveform. So for example, an LFO might be set to 4.47 cycles/second, or in more modern terminology 4.47 Hz.

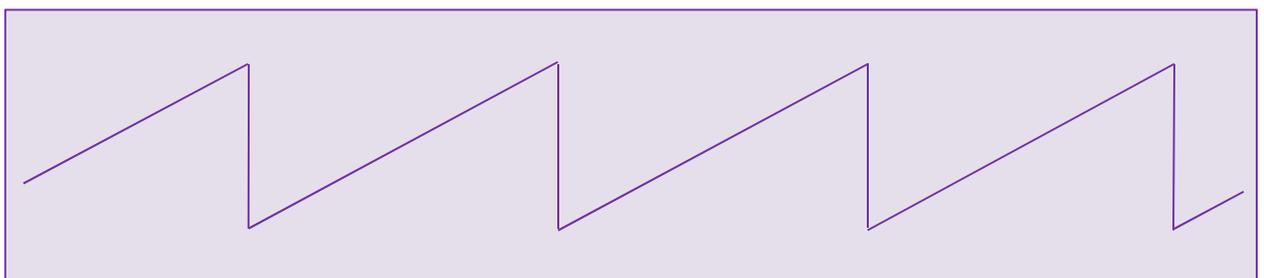
The Triangle Wave

This is a ramp up followed by a ramp down.



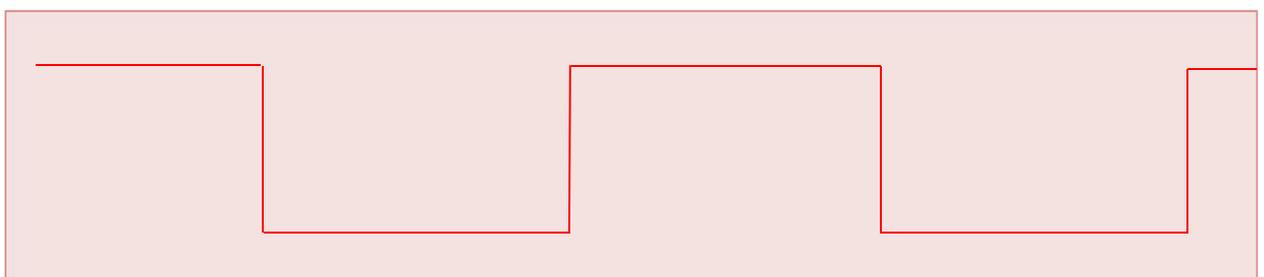
The Sawtooth Wave

This is a ramp up followed by an immediate return to start.



The Square Wave

This alternates between a low and a high value.



Software Implementation

As with the Delay Ramp, software implementations like the Sound Factory operate in completely different ways from their analogue counterparts. A common technique in software for implementing an oscillator is to use a digital count. The following conceptual computer code for a triangle wave illustrates this technique.

	Program Code	Comments
	COUNT = 0	Set the count equal to zero
INC	INCREMENT COUNT	Add 1 to the count
	IF COUNT = MAX THEN DEC ELSE INC	If the count is at maximum then jump to DEC
		If the count is not at maximum then jump back to INC
DEC	DECREMENT COUNT	Subtract 1 from the count
	IF COUNT = MIN THEN INC ELSE DEC	If the count is at minimum then jump to INC
		If the count is not at minimum then jump back to DEC

This code once initiated will run in a continuous loop thereby generating an endlessly cycling waveform. The code for the other types of waveform can be constructed as follows :-

- The code for a sawtooth wave will test for a maximum value, and when detected will set the count back to zero.
- The code for a square wave will repeatedly set a minimum then a maximum value.
- The code for a sine wave is more involved because unlike the other waveforms it is not linear in nature. For this we will need to know what the value of the waveform is at each increment. These values are then stored in what is known as a look-up table, commonly referred to as a Wavetable. To generate the waveform we then cycle continuously through the table reading out each value in turn. This technique is extremely versatile since if we know each of the successive values for any given waveform we can construct any type of oscillator.

Again we can see that modern digital implementations of oscillators are far removed from their original analogue realisations, but the principles and application are the same.

³ [Modulation](#)

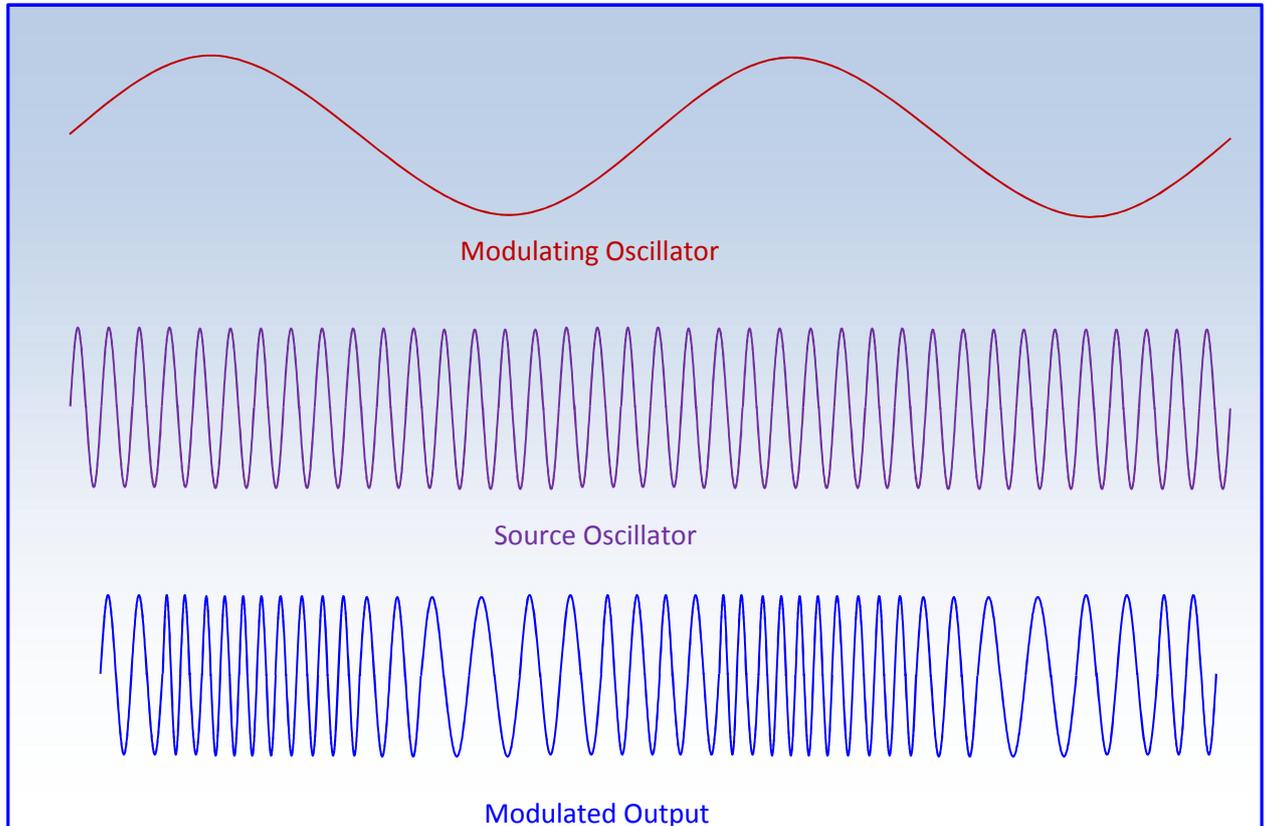
The term Modulation describes a process whereby one oscillator is used to vary the characteristics of another. In analogue synthesisers this technique can be employed in a number of ways, for example to change the pitch, cut off frequency (tone) or volume of a sound. In our Sound Factory example we are using a Low Frequency Oscillator to modulate the pitch of a sound sample and so produce a vibrato. In such an arrangement we can call the LFO the Modulating Oscillator and the sound sample the Source Oscillator.

Analogue Implementation

In an analogue synthesis, a common technique for achieving this effect is to use a particular type of Source Oscillator known as a Voltage Controlled Oscillator (VCO). It is characterised by having a dedicated input to which a control voltage can be applied. The value of this voltage determines the frequency of oscillation. So if we connect the output of an LFO to this control input we can obtain a variation in frequency and therefore in pitch. This is shown diagrammatically below.



For the sake of simplicity let us assume that the Source Oscillator is generating a sine wave. The waveforms that we would then see in this process are as shown below.



We can see from these waveforms that the frequency of the modulated output is now varying in synchronism with the Modulating Oscillator. In particular, when the Modulating Oscillator goes positive we supply more control voltage to the Source Oscillator and its frequency increases. Conversely when the Modulating Oscillator goes negative we supply less control voltage to the Source Oscillator and its frequency decreases. We therefore get a continuous upwards and downwards variation in pitch, and this is our vibrato.

Two more parameters are relevant in this process, the frequency and the amplitude of the Modulating Oscillator. Its frequency determines the overall rate at which the Source Oscillator is being modulated, and therefore the speed of the vibrato. Its amplitude determines the overall pitch variation of the Source Oscillator and therefore the depth of the vibrato. For LFO3 in the Sound Factory we set the former in the LFO3 'Rate Hz' box at the bottom of the Matrix page, and the latter in the Pitch 'Depth' box of the relevant Matrix row.

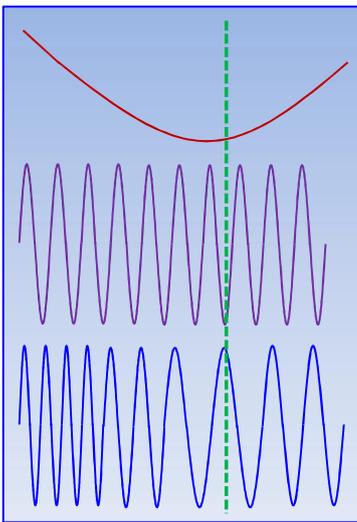
An LFO can generate a number of other types of waveform, so it's interesting to observe what the effect of these would be on the resulting vibrato.

- A triangle wave would vary the frequency smoothly above and below the pitch in a similar manner to the sine wave, but in a more linear fashion. We would be unlikely to notice the difference.

- A sawtooth wave would produce an uneven variation in the vibrato. It would vary the frequency smoothly upwards during the ramp thereby producing a gradual increase in pitch, then there would be a sudden drop in pitch as it returns to start.
- A square wave will produce two sudden changes in frequency, one above pitch then one below. Musically we would perceive this as a trill.

Software Implementation

In software implementations like the Sound Factory the equivalent of the Source Oscillator is the sound sample, and this is just a set of digital numbers. Specifically there is a binary value for every point in time on its waveform. So if we wish to change the waveform in any way it will be necessary to modify these values, a process known as Resampling. For example, to change the volume of the sample we can add or subtract the same fixed number from every value depending on whether the volume is to be increased or decreased.



To modify the frequency of the sample in a manner required for pitch modulation we need to redefine each value on the waveform.

The diagram opposite shows a section of the modulator, source and modulated waveforms previously depicted. At any given point in time, indicated by the green dashed line, we can see how the original sample value needs to be redefined to produce the new modulated value.

To perform this transformation across the entire waveform we need to construct program code that will determine the respective values of the modulating and source waveforms at each point in time. A prescribed set of rules can then use these to compute a new set of values for the modulated waveform.