Music 270a: Waveshaping Synthesis

Tamara Smyth, trsmyth@ucsd.edu
Department of Music,
University of California, San Diego (UCSD)

February 25, 2019

Waveshaping Synthesis

- In waveshaping, it is possible to change the spectrum with the amplitude of the sound (i.e. changing the time-domain waveform by a *controlled* distortion of the amplitude).
- Since this is also a characteristic of acoustic instruments, waveshaping has been used effectively for synthesizing traditional musical instruments, and in particular, brass tones.
- Like FM, waveshaping synthesis enables us to vary the bandwidth and spectrum of a tone in a way that is more computationally efficient than additive synthesis.
- Also like FM, waveshaping provides a continuous control of the spectrum over time by means of an index.
- Unlike FM, waveshaping allows you to create a band-limited spectrum with a specified maximum harmonic number (i.e. making it easier to prevent aliasing!).

Waveshaper

- In a simple waveshaping instrument, an input signal $x(t)$ is passed through a *box* containing a waveshaping function or *transfer function*, also known as a *waveshaper*, $w(x)$.

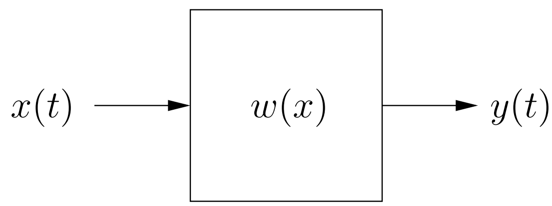

Figure 1: A simple waveshaping instrument with a waveshaping transfer function $w(x)$.

- The transfer function $w(x)$ is typically *nonlinear*, and alters the shape of the input $x(t)$ to produce an output $y(t)$.
- The output, $y(t)$ will depend on:
  1. the nature of the transfer function (the nature of the nonlinearity)
  2. the amplitude of the input signal $x(t)$, e.g., increasing the amplitude of the input may cause the output waveform to change shape.

Indexing

- The transfer function may be an algebraic function of input signal $x(t)$.
- To reduce computation, or to use a waveshaping function that can't be expressed algebraically (e.g. hand-drawn, or data derived elsewhere), the transfer function $w(x)$ may be saved as a vector, or table.
- The waveshaping table $w(x)$ is indexed with the input samples given $x(t)$. This will require
  1. **scaling** $x(t)$, typically between -1 and 1, so that it's peak-to-peak amplitude equals the length of $w(x)$.
  2. **offsetting** the values of $x(t)$ so they are positive and begin with one (1) (since we are using Matlab) so we have positive integers as indices to the table.
  3. **interpolating** the values of $w(x)$ when the index given by $x(t)$ is not an integer.

Linear Interpolation

- Rather than rounding values of $x(t)$ to nearest integers, it is more accurate to *interpolate* between two neighboring values of the wavetable.
- If $x = 6.5$, we could take values from table $w(x)$ at index 6 and 7, and "construct a line between them", i.e., take the value halfway between its neighbours.
- At $x = 6.9749$, we would give greater weight to the 7^{th} element.

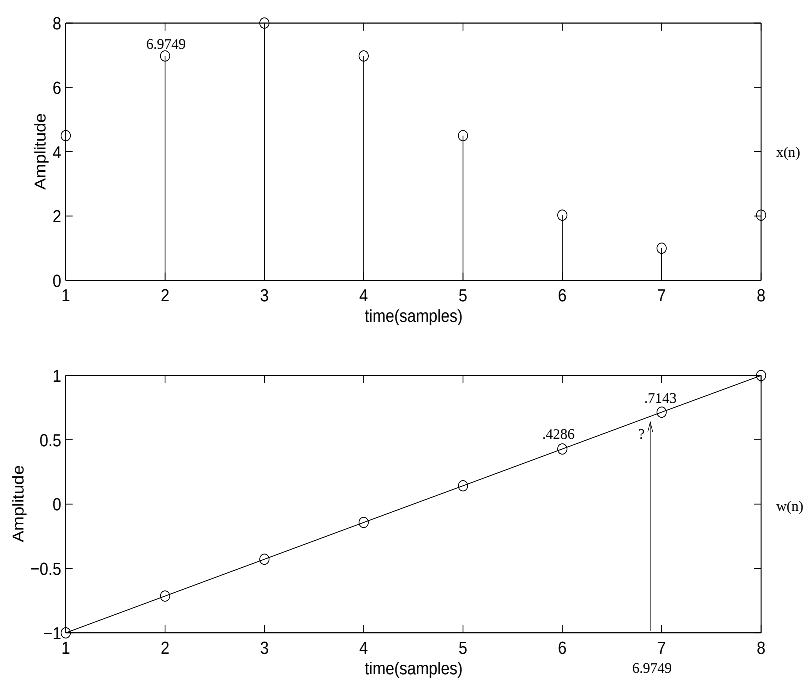

Figure 2: Linear interpolation.

Matlab Linear Interpolation

- More generally, linear interpolation is given by

$$w(n+\eta) = (1-\eta)w(n) + (\eta)w(n+1)$$

where n is the integer part of the original index value, and η is the fractional part, indicating how far from n we want to interpolate,

$$\eta = x - n.$$

- Below is a Matlab function which implements linear interpolation.

```
function y = lininterp(w, x);
% LININTERP Linear interpolation.
%   Y = LININTERP(W, X) where Y is the output,
%   X is the input indeces, not necessarily
%   integers, and W is the transfer function
%   indexed by X.

n = floor(x);
eta = x-n;
w = [w 0];
y = (1-eta).*w(n) + eta.*w(n+1);
y = y(1:length(x));
```

Thru Box

- With a thru box, we define a waveshaping transfer function that will do nothing to the signal.
- What is the shape of such a transfer function?
- Though this may not seem very interesting, it's a good first step in understanding of how we use our waveshaping function and also to make sure we've properly implemented linear interpolation.

```
fs = 8;
dur = 1;
nT = [0:1/fs:dur-1/fs];
N = length(nT);
x = cos(2*pi*(1/dur)*nT);     % input
xsc = (x + abs(min(x)));      % offset x
xsc = xsc/max(xsc)*(N-1) + 1; % scale x
w = linspace(-1, 1, N);       % waveshaper
y = lininterp(w, xsc);
```

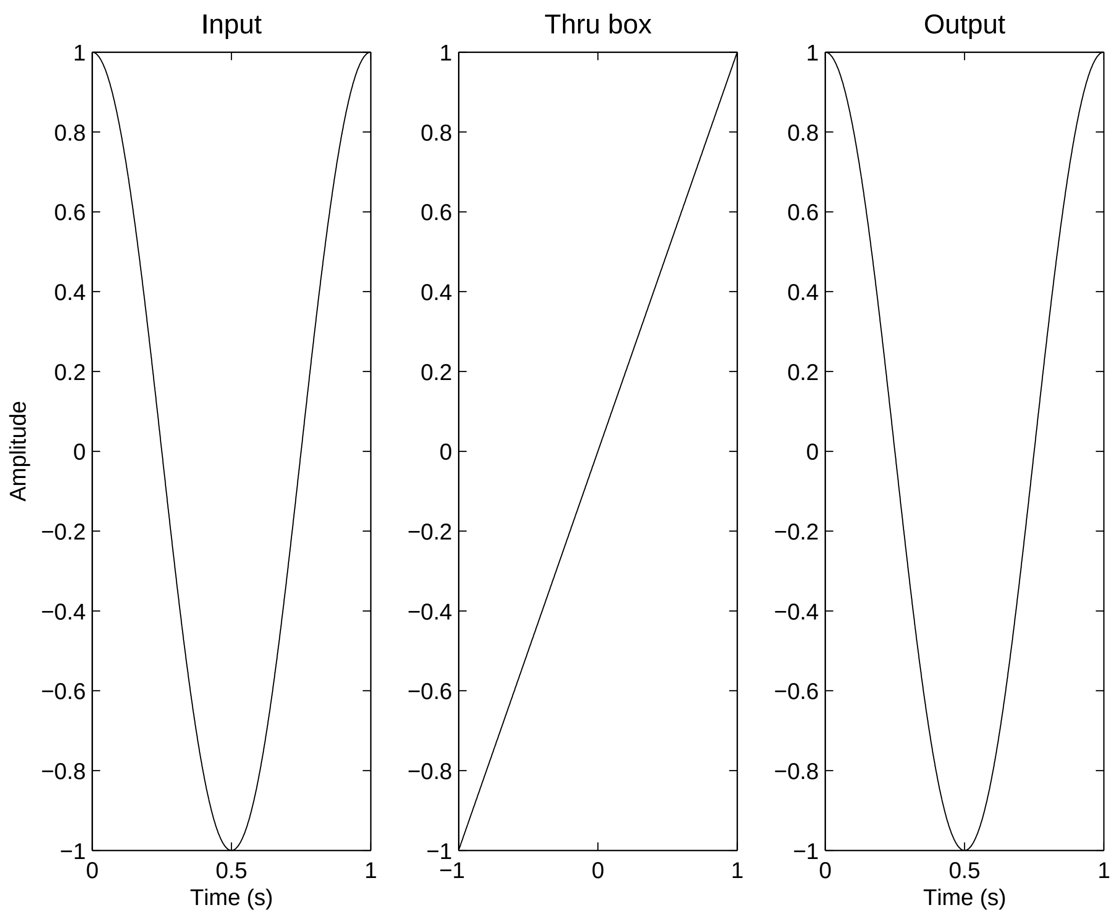

Figure 3: Thru box.

Inverting Box

- Changing the direction of our linear function, we get a waveshaping function that inverts the signal.

```
...
w = linspace(1, -1, N);        % waveshaper
y = lininterp(w, xsc);
```

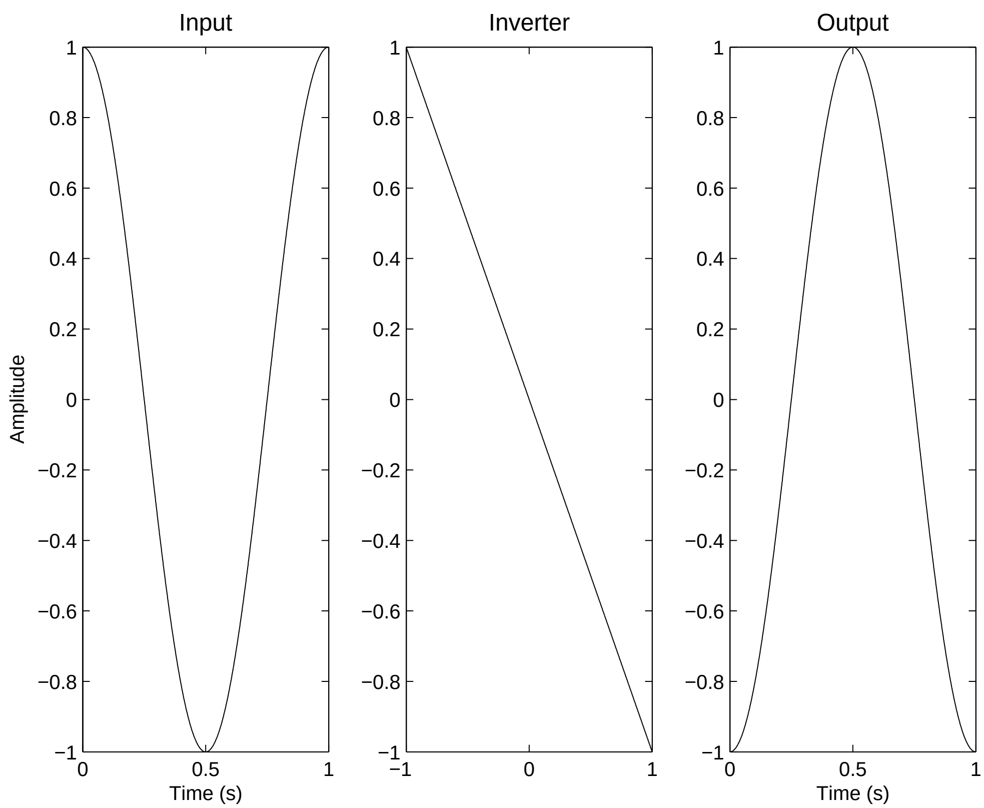

Figure 4: Inverter.

Attenuator Box

- We can also make an attenuator by changing the slope of our linear function.

```
...
m = 0.8;
w = m*linspace(-1, 11, N);   % change slope.
y = lininterp(w, xsc);
```

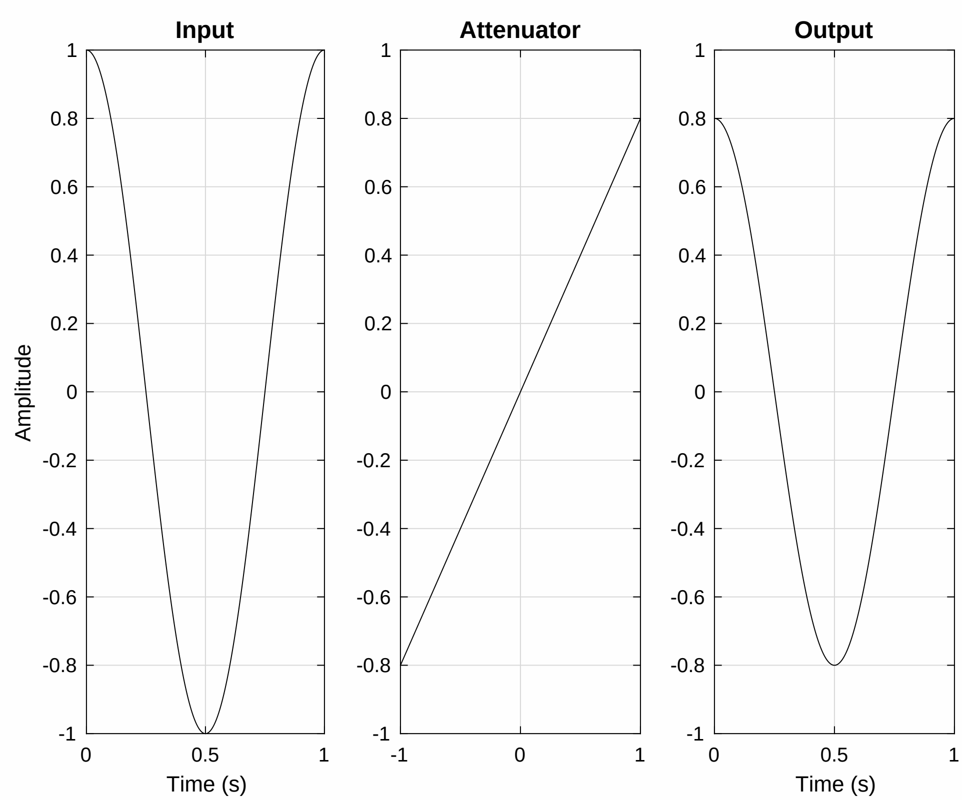

Figure 5: Attenuator (slope of .08)

Transfer Function

- A waveshaper is characterized by its *transfer function* which relates the input signal to the output signal, that is, the output is a function of the input.
- It is represented graphically with the input on the x-axis and the output on the the y-axis.

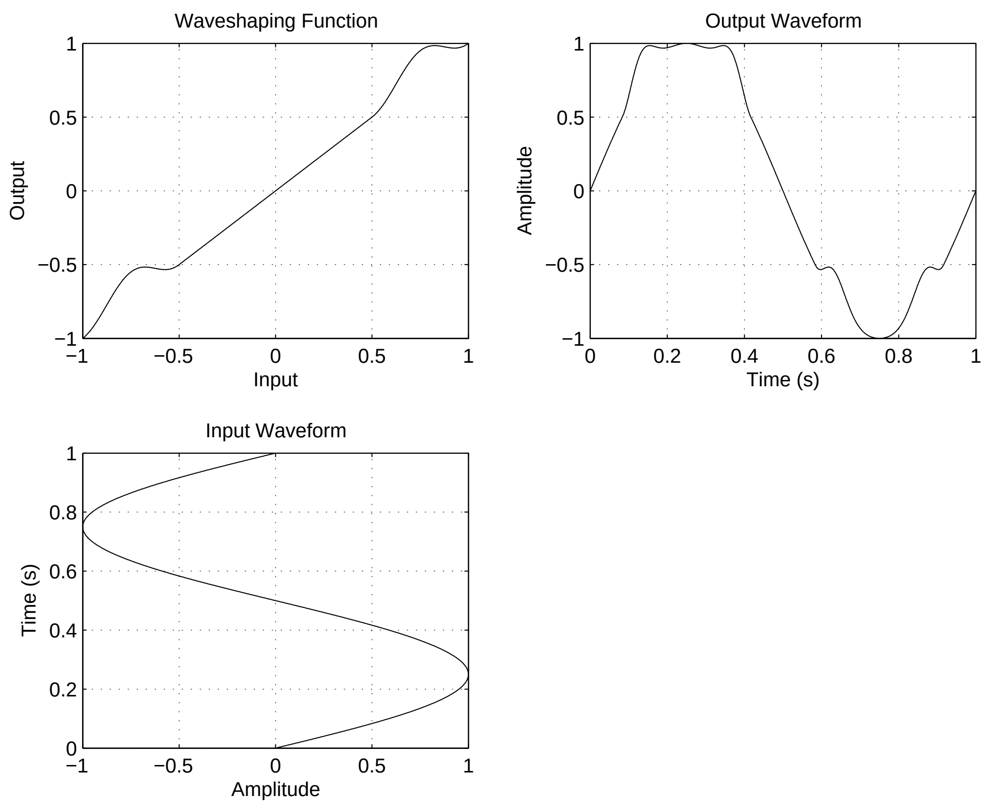

Figure 6: An example *waveshaper* transfer function. The output is determined by the value of the transfer function with respect to the input.

Waveshaper Output

- Notice, in this case, that the shape of the output waveform, and thus the spectrum, changes with the amplitude of the input signal.
- The spectrum becomes richer as the input level is increased, a characteristic we already observed in sounds produced by musical instruments.

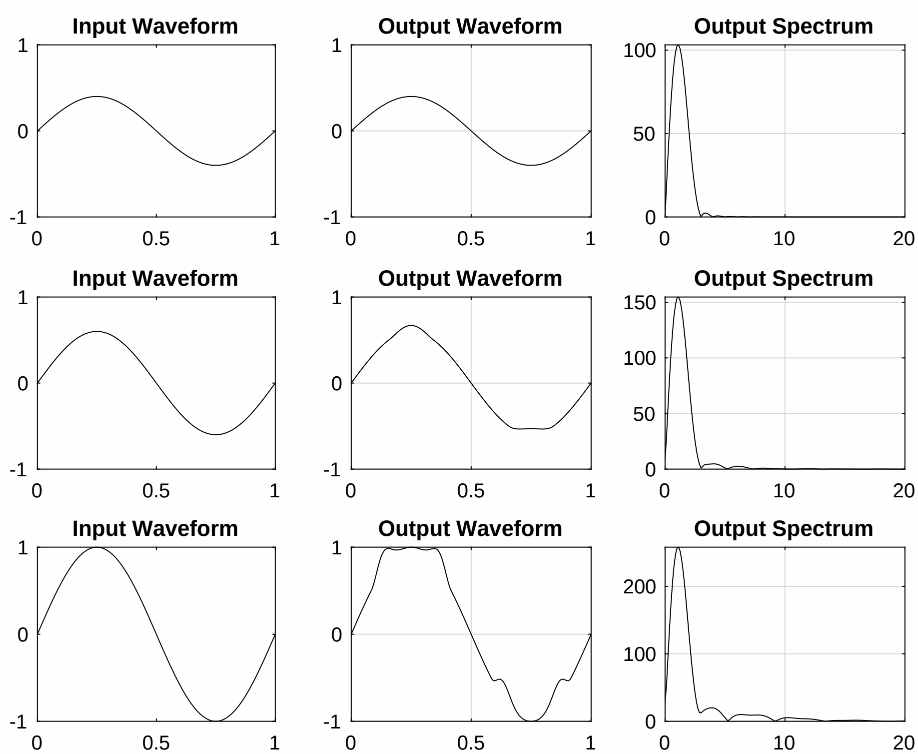

Figure 7: Using the waveshaper from Figure 6, the spectrum becomes richer as the input level is increased.

Even and Odd Transfer Function

- When the transfer function is an odd function[1], the spectrum contains only odd-numbered harmonics.
- When the transfer function is even[2], the spectrum contains only even-numbered harmonics, thereby doubling the fundamental frequency and raising the pitch of the sound by an octave.

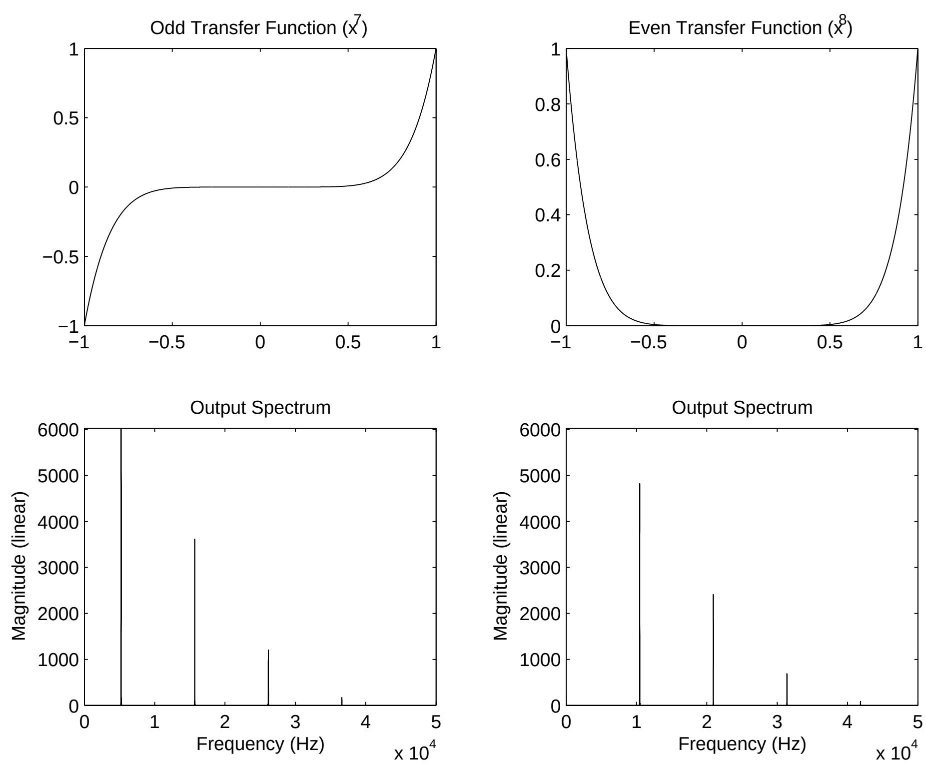

Figure 8: Output Spectrum of even and odd Transfer Functions.

[1]A function $f(n)$ is said to be odd if $f(-n) = -f(n)$.
[2]A function $f(n)$ is said to be even if $f(-n) = f(n)$

Controlling the spectrum

- A waveshaper with a linear transfer function will not produce distortion, but any deviation from a line will introduce some sort of distortion and change the spectrum of the input.
- To control the maximum harmonic in the spectrum (say, for the purpose of avoiding aliasing), a transfer function is expressed as a polynomial:

$$F(x) = d_0 + d_1 x + d_2 x^2 + ... + d_N x^N$$

  where the order of the polynomial is N, and d_i are the polynomial coefficients.
- When driven with a sinusoid, a waveshaper with a transfer function of order N produces no harmonics above the N^{th} harmonic.
- When the driving sinusoid is of unit amplitude, the amplitudes of the various harmonics can be calculated using the right side of Pascal's triangle.

Constructing Pascal's Triangle

- In order to see the amplitudes of the harmonics produced by a term in the polynomial, we can look at Pascal's triangle.
- To construct Pascal's triangle, first create a $N \times N$ table, and input ones along the diagonal, starting from the top left-hand corner (i.e., create an N by N identity matrix).

DIV		h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}
	x^0	1											
	x^1		1										
	x^2			1									
	x^3				1								
	x^4					1							
	x^5						1						
	x^6							1					
	x^7								1				
	x^8									1			
	x^9										1		
	x^{10}											1	
	x^{11}												1

- The symbols along the side, x^j, represent the term in the polynomial.
- The symbols along the top, h_j, represent amplitude of the j^{th} harmonic.

Filling in Pascal's Triangle

- To fill in the values follow the following two steps:
  1. Set a value in column h_0 to twice the value of h_1 from the previous row.
  2. Add two adjacent numbers in the same row and place the sum below the space between them, on the next row.

DIV		h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}
	x^0	1											
	x^1		1										
	x^2	2		1									
	x^3		3		1								
	x^4			4		1							
	x^5				5		1						
	x^6					6		1					
	x^7						7		1				
	x^8							8		1			
	x^9								9		1		
	x^{10}									10		1	
	x^{11}										11		1

- Continue these two steps, first by filling in the next value for h_0, and then taking the adjacent sum.

- Finally, to obtain the divider DIV, multiply the value of DIV from the previous row by 2, starting in row x^0 with 0.5.

DIV		h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}
0.5	x^0	1											
1	x^1	0	1										
2	x^2	2	0	1									
4	x^3	0	3	0	1								
8	x^4	6	0	4	0	1							
16	x^5	0	10	0	5	0	1						
32	x^6	20	0	15	0	6	0	1					
64	x^7	0	35	0	21	0	7	0	1				
128	x^8	70	0	56	0	28	0	8	0	1			
256	x^9	0	126	0	84	0	36	0	9	0	1		
512	x^{10}	252	0	210	0	120	0	45	0	10	0	1	
1024	x^{11}	0	462	0	330	0	165	0	55	0	11	0	1

Calculating Spectral Output

- Notice from Pascal's triangle that if the order of the polynomial is even, only even harmonics will be present.
- If the order is odd, only odd harmonics will be present.
- If the transfer function $F(x) = x^5$ is driven by an oscillator of amplitude 1, the output will contain the first, third and fifth harmonics with the following amplitudes:

$$\begin{aligned} h_1 &= \frac{1}{16}(10) = 0.625 \\ h_3 &= \frac{1}{16}(5) = 0.3125 \\ h_5 &= \frac{1}{16}(1) = 0.0625 \end{aligned}$$

Transfer function $F(x) = x^5$

- Create a 1 second long 220 Hz sinusoid input x and plot the output $y = x^5$ in Matlab:

```
fs = 44100;
nT = 0:1/fs:1;
x = sin(2*pi*220*nT);
y = x.^5;
```

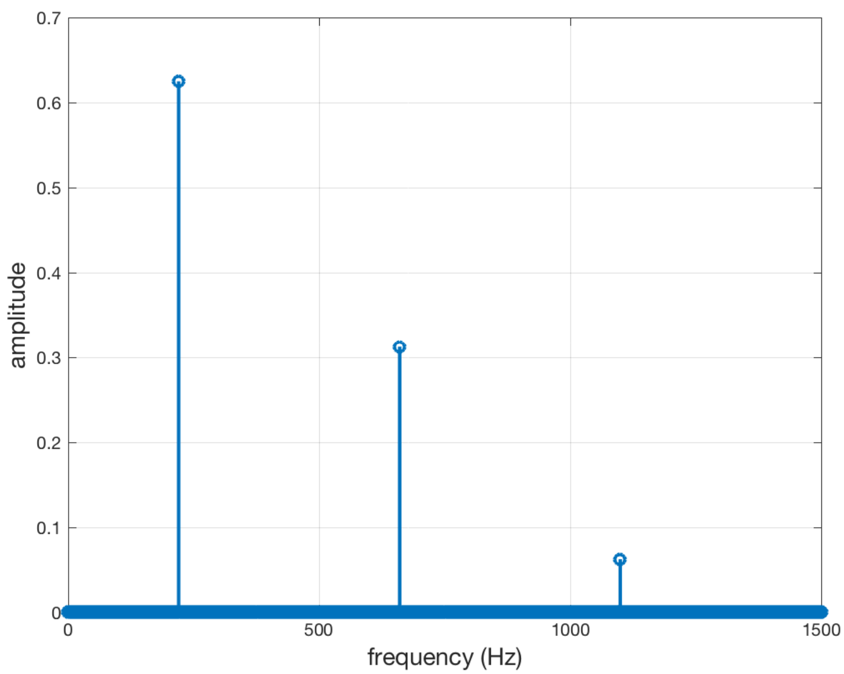

Figure 9: Output spectrum of the transfer function $y = x^5$, where x is a unit amplitude sinusoid at a frequency of 220 Hz.

Transfer Functions with Multiple Terms

- If the transfer function has multiple terms, then the output will be the sum of the contributions of each term.
- For example, the transfer function

$$F(x) = x + x^2 + x^3 + x^4 + x^5$$

produces an output spectrum with the following harmonic amplitudes:

$$\begin{aligned}
h_0 &= \frac{1}{2}(2) + \frac{1}{8}(6) = 1.75 \\
h_1 &= 1 + \frac{1}{4}(3) + \frac{1}{16}(10) = 2.375 \\
h_2 &= \frac{1}{2}(1) + \frac{1}{8}(4) = 1.0 \\
h_3 &= \frac{1}{4}(1) + \frac{1}{16}(5) = 0.5625 \\
h_4 &= \frac{1}{8}(1) = 0.125 \\
h_5 &= \frac{1}{16}(1) = 0.0625
\end{aligned}$$

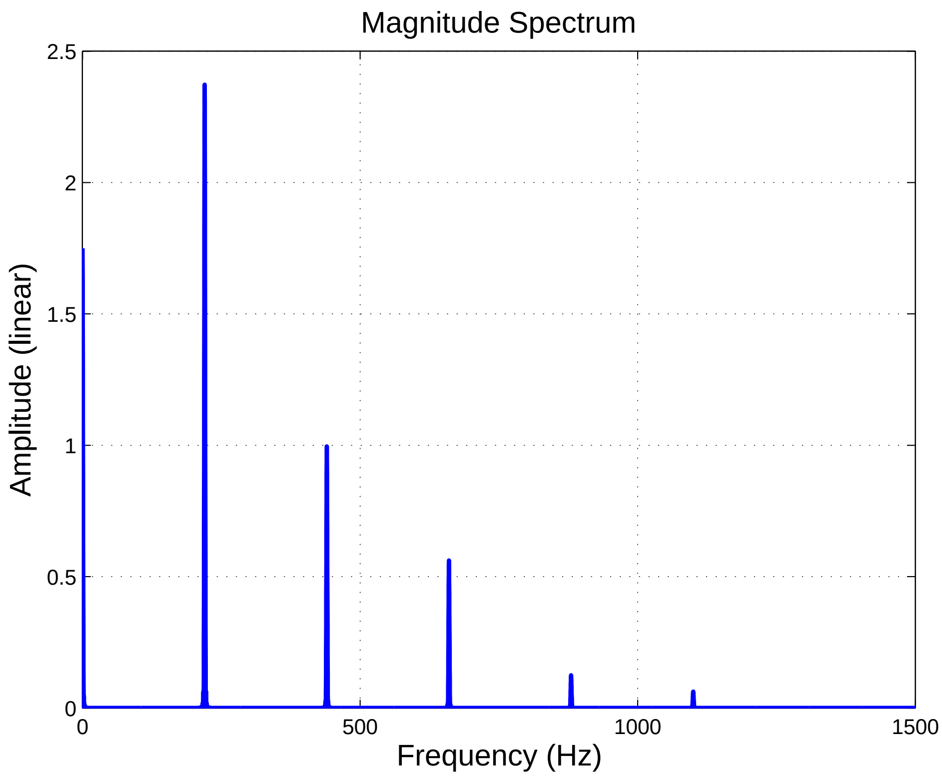

Figure 10: Output spectrum of the transfer function $y = x + x^2 + x^3 + x^4 + x^5$, where x is a unit amplitude sinusoid at a frequency of 220 Hz.

Non-sinusoidal input

- The previous calculations are based on a unit-amplitude sinusoidal input.
- Non sinusoidal input to the waveshapping function produces less predictable output, and therefore is more difficult to keep alias free.
- It is, however, possible to change the amplitude of the sinusoidal input so that it is less than—or greater than—1.
- This creates a distortion index similar to the modulation index seen in FM synthesis.

Distortion Index

- If the input cosine has an amplitude of a, then the output in polynomial form becomes
$$F(ax) = d_0 + d_1 ax + d_2 a^2 x^2 + ... + d_N a^N x^N$$
- **Example:** Given the waveshapping transfer function
$$F(x) = x + x^3 + x^5,$$
an input sinusoid with amplitude a yields the output
$$F(ax) = ax + (ax)^2 + (ax)^5,$$
with the amplitude of each harmonic calculated using Pascal's triangle to obtain
$$\begin{aligned} h_1(a) &= a + \frac{1}{4}3a^3 + \frac{1}{16}10a^5 \\ h_3(a) &= \frac{1}{4}a^3 + \frac{1}{16}5a^5 \\ h_5(a) &= \frac{1}{16}a^5 \end{aligned}$$
- Because an increase in a (typically between 0 and 1) produces a richer output spectrum, it is often referred to as a *distortion index* (analogous to the index of modulation in FM synthesis).

Selecting a Tranfer Function

- **Spectral Matching**: Select a transfer function that matches a desired steady-state spectrum for a particular distortion index a.
- This may be done using *Chebyshev polynomials* of the first kind, denoted $T_k(x)$, where k is the order of the polynomial.
- The zeroth- and first-order Chebyshev polynomials are given by

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \end{aligned}$$

  and higher-order polynomials are given by

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x).$$

- These polynomials have the property that when a **sinusoid of unit amplitude** is applied to the input, the output signal contains only the k^{th} harmonic.

The first few Chebyshev Polynomials of the first kind

- For your convenience, here are some of the first few:

$$
\begin{aligned}
T_0(k) &= 1 \\
T_1(k) &= x \\
T_2(k) &= 2x^2 - 1 \\
T_3(k) &= 4x^3 - 3x \\
T_4(k) &= 8x^4 - 8x^2 + 1 \\
T_5(k) &= 16x^5 - 20x^3 + 5x
\end{aligned}
$$

- The rest may be generated in Matlab using the following:

```
T(:, 1) = ones(length(x), 1);
T(:, 2) = x;
for n = 3:Hmax+1
  T(:, n) = 2*x.*T(:,n-1) - T(:,n-2);
end
```

Matching a Spectrum Using Chebyshev Polynomials

- A spectrum containing several harmonics can be matched by combining the appropriate Chebyshev polynomial for each harmonic into a single transfer function.
- Let h_j be the amplitude of the j^{th} harmonic, and N be the highest harmonic in the spectrum. The transfer function is then given by:

  $$F(x) = h_0T_0(x)+h_1T_1(x)+h_2T_2(x)+\cdots+h_NT_N(x).$$

Example of Spectral Matching

- Given the following spectrum, what would be the transfer function?

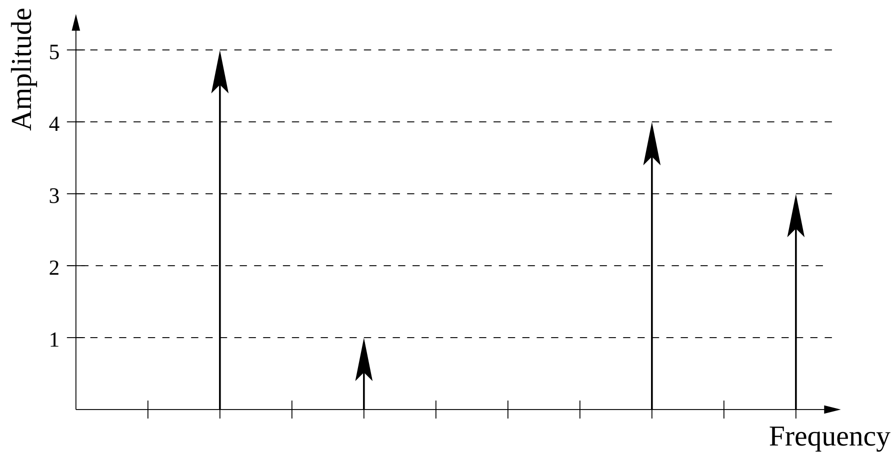

Figure 11: A steady state spectrum.

- The spectrum contains the first, second, fourth, and fifth harmonics, with amplitudes 5, 1, 4, 3, respectively.

- The transfer function is given by

$$
\begin{aligned}
F(x) &= 5T_1(x) + T_2(x) + 4T_4(x) + 3T_5(x) \\
&= 5x + (2x^2 - 1) + 4(8x^4 - 8x^2 + 1) \\
&\quad +3(16x^5 - 20x^3 + 5x) \\
&= 48x^5 + 32x^4 - 60x^3 - 30x^2 + 20x + 3.
\end{aligned}
$$

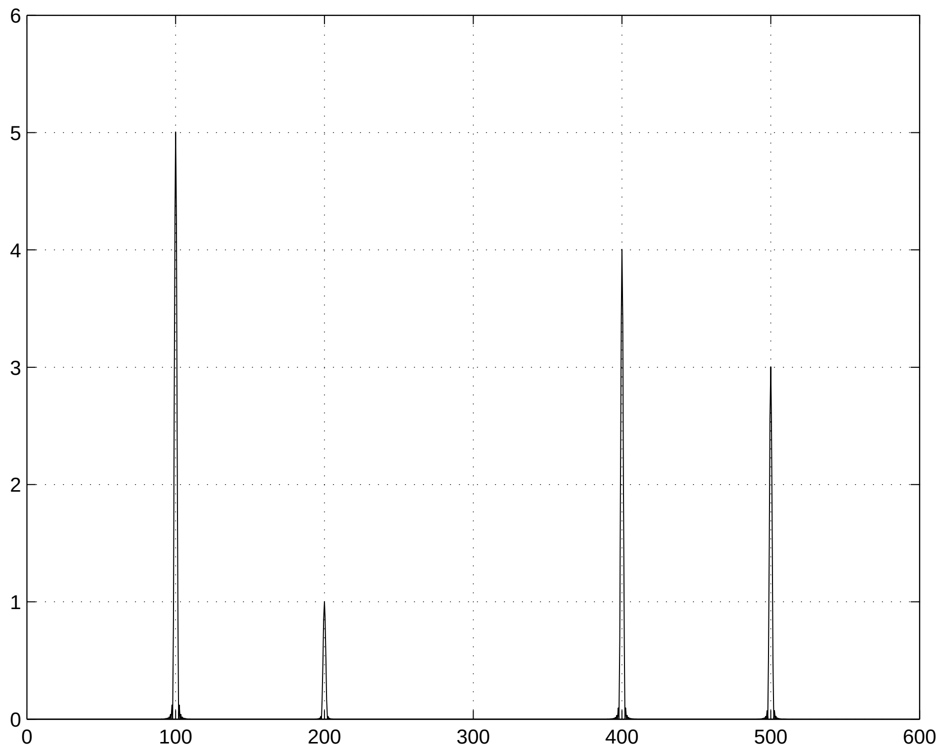

Figure 12: The steady state plotted in Matlab.

Selecting a Polynomial to Fit Data

- If you wish to construct a waveshaper based on incoming data, then you will create a table, and proceed using linear interpolation (as shown in previous slides).
- The problem with this approach is that you can't ensure a bandlimited spectrum withouth aliasing.
- It is also possible to fit a polynomial to the data (there are many ways of doing this, the details go beyond the scope of this class).
- You may like to take advantable of Matlab's `polyfit` to accomplish this task.