

Piecewise Linear Circuits

Kenneth A. Kuhn
March 24, 2004

Introduction

Piecewise linear circuits are used to approximate non-linear functions such as sine, square-root, logarithmic, exponential, etc. The quality of the approximation depends on the number of segments and the strategy used in determining the segments. There are two general approaches to building piecewise linear circuits: (1) non-linear voltage dividers with diodes (or transistors) used to switch between the segments and (2) summing the outputs of a chain of saturating amplifiers. Both of these approaches are technically the same although each has its advantages and disadvantages. The diode approach has the advantage of simplicity but the disadvantages include temperature dependence on the switching thresholds and relatively slow response. The saturating amplifier method has the disadvantage of complexity but the advantages of minimal temperature dependence on thresholds and high speed. The focus for this course will be on diode switching.

The piecewise linear circuits we will discuss are divided into two types: compression and expansion and the circuit topology is different for each type. Compression circuits compress the dynamic range of an input signal. Examples include square-root, logarithmic, and sine. Expansion circuits expand the dynamic range of an input signal and a common example is the exponential function. Sometimes a combination of both types of circuits is needed to implement a particular transfer function such as to create a linear frequency versus voltage tuning characteristic for a varactor tuned oscillator (a varactor is a diode whose junction capacitance can be varied by varying the reverse bias).

The student must learn how to analyze piecewise linear circuits and then how to design the circuits. We will begin with analysis. Since the circuits are non-linear, superposition will not generally apply although it will be valid over each linear segment. It is important to understand when superposition can and can not be used. The starting point should always be with the linear segment for the smallest signal. Then we progress one segment at a time until the last segment handling the largest signals is active. Each segment analysis is simple (i.e. linear) and builds on the result of the previous segment analysis. Our goal is to determine the breakpoints and slopes of each segment.

For design, we will have determined a list of breakpoints and slopes to fit the desired function. This list may have been determined graphically or by some mathematical process. However the list was determined the goal is to implement it in hardware. As in analysis, we begin with the segment that applies to the smallest input signal and progress one segment at a time until we are finished with the last segment handling the largest input signal. Each segment design is simple (i.e. linear) and builds on the results of the previous segment design.

Compression circuits are generally built as non-linear voltage dividers. Expansion circuits are generally built by constructing the inverse compression circuit and using this circuit in the feedback of a non-inverting op-amp. The focus of this paper will be on compression circuits.

Piecewise Linear Circuits

Analysis

The general process of analysis and design can be illustrated by an example. The example will be a simple implementation of a network to approximate the square-root of an input voltage over the range of 1 to 10 volts. This is the same circuit that is designed later. The goal of analysis is to determine the breakpoints and the slopes of the different segments. From this data a plot of the transfer function can be made. The circuit is shown in Figure 1.

The first thing to do is to establish the breakpoints. Analyzing the voltage divider shows that:

$$\begin{aligned}V_A &= 1.69 * 51 / (51 + 68 + 75) = 0.44 \text{ volts} \\V_B &= 1.69 * (51 + 68) / (51 + 68 + 75) = 1.04 \text{ volts} \\V_C &= 1.69 \text{ volts}\end{aligned}$$

Using a reasonable model (discussed below) for the diode of 0.56 volts then gives us the real breakpoints of:

$$\begin{aligned}V_1 &= 0.44 + 0.56 = 1.00 \text{ Volts} \\V_2 &= 1.04 + 0.56 = 1.60 \text{ Volts} \\V_3 &= 1.69 + 0.56 = 2.25 \text{ Volts}\end{aligned}$$

Now we can calculate the slopes by using the voltage divider equation. We note that:

$$\begin{aligned}R_1 &= 6,800 \text{ Ohms} \\R_2 &= 7,190 \text{ Ohms} \\R_3 &= 6,600 \text{ Ohms}\end{aligned}$$

For input voltages below the first breakpoint of 1.00 volts, the slope of the output is 1.00 by definition since no voltage division is taking place. For output voltages between the first and second breakpoints the slope is the voltage division of R_0 and R_1 which is: $6,800 / (10,000 + 6,800) = 0.405$.

For output voltages between the second and third breakpoints the slope is the voltage division of R_0 and $R_1||R_2$ which is $3,495 / (10,000 + 3,495) = 0.259$.

For output voltages higher than the third breakpoint the slope is the voltage division of R_0 and $R_1||R_2||R_3$ which is $2,285 / (10,000 + 2,285) = 0.186$.

For plotting the transfer function we would like to know the input breakpoints. These can be calculated from the above data. The first input breakpoint is at 1.00 volts because up until that point there is no voltage division taking place and the output simply equals the input. For the output breakpoint of 1.60 Volts, the input breakpoint is found by the general equation:

$$X_a + (Y_b - Y_a) / \text{SLOPE}_{ba} = X_b$$

Piecewise Linear Circuits

Where X_a is the previous breakpoint, X_b is the next X breakpoint (the one we are looking for), Y_b is the Y value of the breakpoint we know, and $SLOPE_{ba}$ is the slope of the segment from point a to point b.

Substituting our data give:

$$1.0 + (1.60 - 1.00) / 0.405 = 2.48 \text{ volts}$$

The next input breakpoint is found by

$$2.48 + (2.25 - 1.60) / 0.259 = 4.99 \text{ volts}$$

There are no more breakpoints but we need an easy way to plot the final line segment for inputs above the last breakpoint. All we have to do is to pick any point we like that is spaced about as wide as practical from the last breakpoint and compute the corresponding input voltage. This is done the same way as in the previous calculations. I will choose 3.00 volts. Thus,

$$4.99 + (3.00 - 2.25) / 0.186 = 9.02 \text{ volts.}$$

Thus, we have the following sequence of x and y data pairs that describes the piecewise linear transfer function.

<u>x</u>	<u>y</u>	
0.00	0.00	
1.00	1.00	
2.48	1.60	
4.99	2.25	
9.02	3.00	note: this line segment continues as far past this point as will fit on the graph.

After plotting this data we are now finished with the analysis.

Piecewise Linear Circuits

Square Root Circuit
over 1.0 to 10 V range

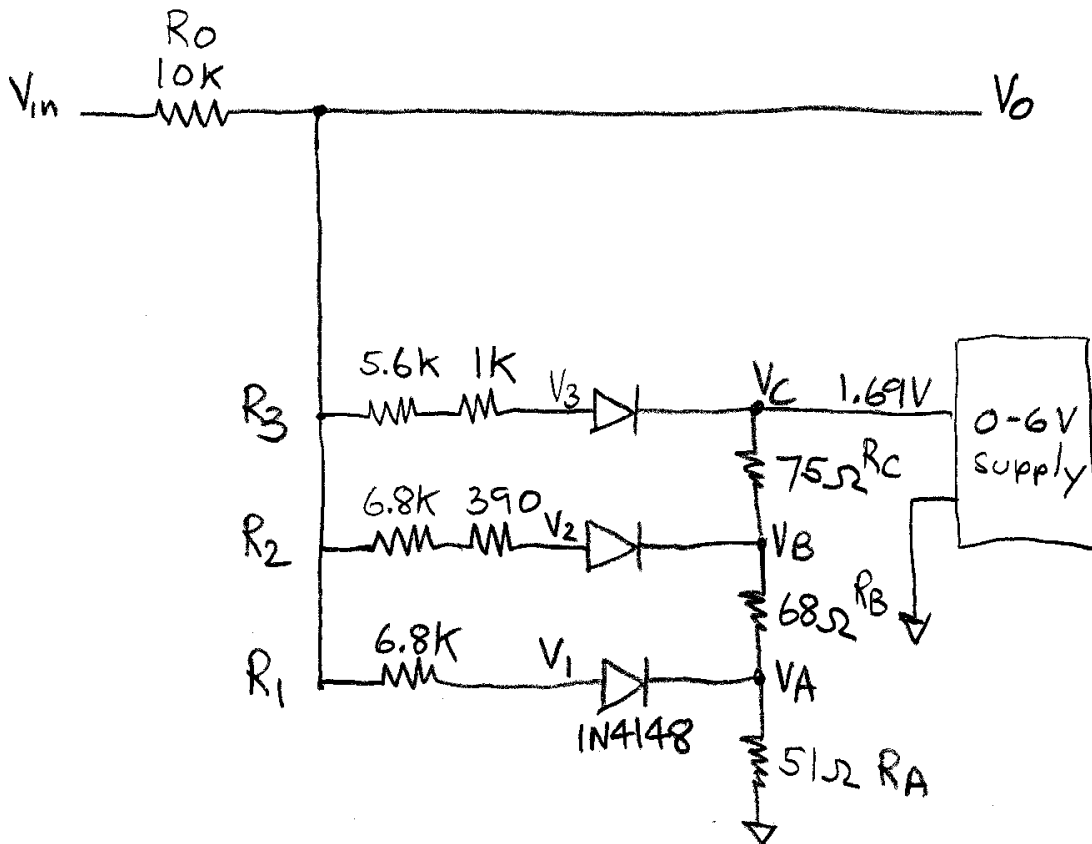
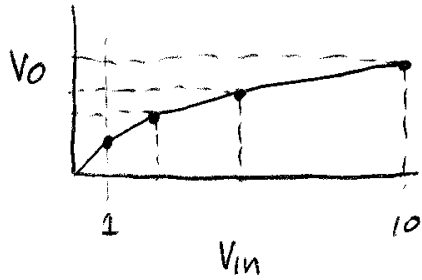


Figure 1: Simple piecewise-linear square-root circuit

Piecewise Linear Circuits

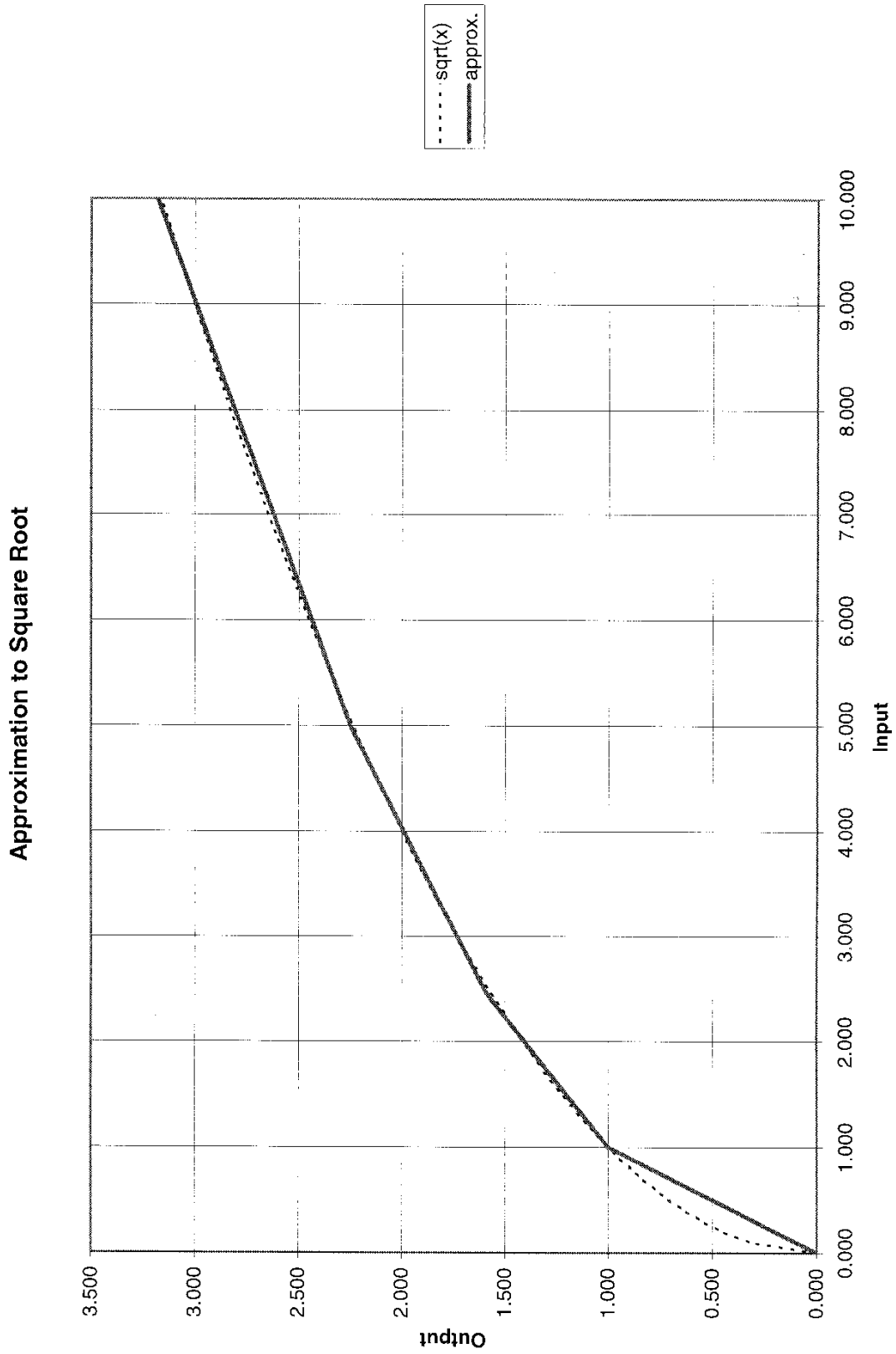


Figure 2: Plot of piecewise-linear square-root circuit

Piecewise Linear Circuits

Design

The design of a square-root network will now be illustrated. The first thing to do is to establish the signal range over which the square-root will be computed (I will specify this as 1 to 10 Volts). The next thing to do is to construct a spreadsheet to assist in fitting a series of line segments to the exact function. The number of required segments is then determined based on the desired accuracy. Our goal is to determine the minimum number of segments that will meet an accuracy specification (for this example we will use about 1 percent).

The spreadsheet is first built to plot the true curve representing the desired function. Then straight lines are manually drawn to fit the curve with minimum error. These lines are approximate and only a starting point to serve as a basis for constructing the lines in the spreadsheet. In general the end points of the segments will not fall exactly on the true curve except possibly at the beginning or end of the function – this results in a more accurate approximation – but is more difficult for the human to work with as this introduces many degrees of freedom to optimize. The spacing of the points is never linear. The rough optimal spacing is inversely related to the magnitude of the second derivative of the function. That is – more segments are used where the slope is changing. I determined that three line segments over the 1 to 10 volt range can approximate the square-root with less than 1 percent error except at just a few areas (the maximum error in this example is 1.39 percent – more refinement in the line segments might reduce the error).

You will observe that the line segments are specified in the spreadsheet as a set of x and y points. The resulting $y = mx + b$ equations are computed by the spreadsheet for the data points. The columns of data represent the input, the true value, and the approximate value determined by the line segments. The next columns are the error (approximate – exact), and error squared. At the top of the error squared column is the sum of the square of the individual data. The goal is to manipulate the x,y data points to minimize this term. Once the term has been reduced about as far as practical then the resulting data and line equations are used to design the network. The equation below is from a cell in the approximate column to illustrate how to automatically select the right line equation.

```
=IF(A13<$A$6,A13*$E$6+$F$6,IF(A13<$A$7,A13*$E$7+$F$7,IF(A13<$A$8,A13*$E$8+$F$8,A13*$E$9+$F$9)))
```

The line segments are plotted on top of the exact curve as a visual guide. The break points and slopes are:

Breakpoint	Slope
1.00	0.407
1.59	0.259
2.25	0.186

We do not implement the last point (3.18) as that point is only used to establish the line. In general we never implement the last point.

Piecewise Linear Circuits

Spreadsheet for approximating the square root by straight line segments

Data points:

<u>x</u>	<u>y</u>
0	0
1	1
2.45	1.59
5	2.25
10	3.18

Line segment calculations:

<u>m</u>	<u>b</u>
1.000	0.000
0.407	0.593
0.259	0.956
0.186	1.320

<u>x</u>	<u>sqrt(x)</u>	<u>approx.</u>	<u>error</u>	<u>error^2</u>
0.000	0.000	0.000		
0.100	0.316	0.100		
0.200	0.447	0.200		
0.300	0.548	0.300		
0.400	0.632	0.400		
0.500	0.707	0.500		
0.600	0.775	0.600		
0.700	0.837	0.700		
0.800	0.894	0.800		
0.900	0.949	0.900		
1.000	1.000	1.000	0.000	0.000000
1.100	1.049	1.041	-0.008	0.000066
1.200	1.095	1.081	-0.014	0.000198
1.300	1.140	1.122	-0.018	0.000328
1.400	1.183	1.163	-0.020	0.000419
1.500	1.225	1.203	-0.021	0.000454
1.600	1.265	1.244	-0.021	0.000432
1.700	1.304	1.285	-0.019	0.000361
1.800	1.342	1.326	-0.016	0.000260
1.900	1.378	1.366	-0.012	0.000149
2.000	1.414	1.407	-0.007	0.000054
2.100	1.449	1.448	-0.002	0.000002
2.200	1.483	1.488	0.005	0.000025
2.300	1.517	1.529	0.012	0.000154
2.400	1.549	1.570	0.020	0.000419
2.500	1.581	1.603	0.022	0.000475
2.600	1.612	1.629	0.016	0.000268
2.700	1.643	1.655	0.012	0.000133
2.800	1.673	1.681	0.007	0.000053
2.900	1.703	1.706	0.004	0.000012
3.000	1.732	1.732	0.000	0.000000
3.100	1.761	1.758	-0.002	0.000006
3.200	1.789	1.784	-0.005	0.000022
3.300	1.817	1.810	-0.007	0.000043
3.400	1.844	1.836	-0.008	0.000064
3.500	1.871	1.862	-0.009	0.000082
3.600	1.897	1.888	-0.010	0.000094
3.700	1.924	1.914	-0.010	0.000100
3.800	1.949	1.939	-0.010	0.000099
3.900	1.975	1.965	-0.010	0.000091

sum e^2= **0.018** Choose points to minimize this number

Only data points from 1.0 are used for error calculations

Figure 3: Excel spreadsheet to tweak breakpoints for minimum sum of error²
 note: Figure 3 continues on the next two pages

Piecewise Linear Circuits

4.000	2.000	1.991	-0.009	0.000078
4.100	2.025	2.017	-0.008	0.000061
4.200	2.049	2.043	-0.006	0.000042
4.300	2.074	2.069	-0.005	0.000023
4.400	2.098	2.095	-0.003	0.000008
4.500	2.121	2.121	-0.001	0.000001
4.600	2.145	2.146	0.002	0.000003
4.700	2.168	2.172	0.004	0.000019
4.800	2.191	2.198	0.007	0.000054
4.900	2.214	2.224	0.011	0.000111
5.000	2.236	2.250	0.014	0.000194
5.100	2.258	2.269	0.010	0.000106
5.200	2.280	2.287	0.007	0.000047
5.300	2.302	2.306	0.004	0.000013
5.400	2.324	2.324	0.001	0.000000
5.500	2.345	2.343	-0.002	0.000005
5.600	2.366	2.362	-0.005	0.000023
5.700	2.387	2.380	-0.007	0.000053
5.800	2.408	2.399	-0.010	0.000091
5.900	2.429	2.417	-0.012	0.000134
6.000	2.449	2.436	-0.013	0.000182
6.100	2.470	2.455	-0.015	0.000232
6.200	2.490	2.473	-0.017	0.000282
6.300	2.510	2.492	-0.018	0.000331
6.400	2.530	2.510	-0.019	0.000377
6.500	2.550	2.529	-0.021	0.000421
6.600	2.569	2.548	-0.021	0.000460
6.700	2.588	2.566	-0.022	0.000494
6.800	2.608	2.585	-0.023	0.000524
6.900	2.627	2.603	-0.023	0.000547
7.000	2.646	2.622	-0.024	0.000564
7.100	2.665	2.641	-0.024	0.000575
7.200	2.683	2.659	-0.024	0.000580
7.300	2.702	2.678	-0.024	0.000578
7.400	2.720	2.696	-0.024	0.000571
7.500	2.739	2.715	-0.024	0.000558
7.600	2.757	2.734	-0.023	0.000539
7.700	2.775	2.752	-0.023	0.000515
7.800	2.793	2.771	-0.022	0.000486
7.900	2.811	2.789	-0.021	0.000453
8.000	2.828	2.808	-0.020	0.000417
8.100	2.846	2.827	-0.019	0.000378
8.200	2.864	2.845	-0.018	0.000337
8.300	2.881	2.864	-0.017	0.000295
8.400	2.898	2.882	-0.016	0.000252
8.500	2.915	2.901	-0.014	0.000210
8.600	2.933	2.920	-0.013	0.000168
8.700	2.950	2.938	-0.011	0.000129
8.800	2.966	2.957	-0.010	0.000094
8.900	2.983	2.975	-0.008	0.000062
9.000	3.000	2.994	-0.006	0.000036
9.100	3.017	3.013	-0.004	0.000016

Piecewise Linear Circuits

9.200	3.033	3.031	-0.002	0.000004
9.300	3.050	3.050	0.000	0.000000
9.400	3.066	3.068	0.002	0.000006
9.500	3.082	3.087	0.005	0.000023
9.600	3.098	3.106	0.007	0.000052
9.700	3.114	3.124	0.010	0.000094
9.800	3.130	3.143	0.012	0.000151
9.900	3.146	3.161	0.015	0.000224
10.000	3.162	3.180	0.018	0.000314

Figure 3 concludes

We are now ready to determine the resistor values. Our entire design consists of designing simple voltage dividers. So, let us review the equation for a voltage divider formed by an upper resistor, R_1 , and a lower resistor, R_2 .

Analysis equation:

$$k = R_2 / (R_1 + R_2) = 1 / [(R_1 / R_2) + 1]$$

Design equation:

$$(R_1 / R_2) = (1 / k) - 1$$

Note that k is the slope. We note that for input signals below the first breakpoint (1.00 volts) that the output voltage equals the input as all of the diodes are reversed biased. For input signals between the first and second breakpoint the slope of the output is just that of the simple voltage divider formed by R_0 and R_1 . For input signals between the second and third breakpoints the slope of the output voltage is that of the voltage divider formed by R_0 and the parallel combination of R_1 and R_2 . For input signals above the third breakpoint the slope of the output is that of the voltage divider formed by R_0 and the parallel combination of R_1 , R_2 , and R_3 . This observation leads to a very simple way to calculate the required resistors R_1 , R_2 , and R_3 relative to a chosen value for R_0 .

We will choose the value of R_0 to be 10,000 Ohms for convenience. This is always a choice.

The ratio, $(R_0 / R_1) = (1 / 0.407) - 1 = 1.457$. Thus, $R_1 = 10,000 / 1.457 = 6,863$ Ohms. We will round this value to the nearest standard value of $R_1 = 6,800$ Ohms.

The ratio, $(R_0 / R_1 || R_2) = (1 / 0.259) - 1 = 2.861$. Thus, $R_1 || R_2 = 10,000 / 2.861 = 3,495$ Ohms. Since we already know that R_1 is 6,800 Ohms then we can calculate R_2 as follows using the inverse parallel resistance equation.

$R_2 = (6,800 * 3,495) / (6,800 - 3,495) = 7,191$ Ohms. This value is about in the midway between two standard resistor values so for better accuracy we will use the series combination of 6,800 and 390 Ohms to make 7,190 Ohms for R_2 .

Piecewise Linear Circuits

The ratio, $(R_0 / R_1 \parallel R_2 \parallel R_3) = (1 / 0.186) - 1 = 4.376$. Thus, $R_1 \parallel R_2 \parallel R_3 = 10,000 / 4.376 = 2,285$ Ohms. Using the actual value of $R_1 \parallel R_2 = 6,800 \parallel 7,190 = 3,495$ Ohms, we can calculate R_3 as $(3,495 * 2,285) / (3,495 - 2,285) = 6,600$ Ohms. We will use the series combination of 5,600 and 1,000 Ohms to make R_3 .

Now we design the breakpoint voltage dividers. One complication is that we have to allow for the voltage drop across the diodes. Unfortunately, this voltage drop is temperature dependent and also current dependent. Another issue that we have to consider is the source resistance of the voltage divider which adds to the resistance values computed above. This issue can be corrected by subtracting the source resistance from each computed resistance above.

One model that gives reasonably accurate results for a common 1N4148 diode is to use a fixed voltage drop of 0.56 volts and a series resistance of 33 Ohms. The switching point is not abrupt but is spread over several tenths of a volt beginning at about 0.4 volts. This “soft turn-on” can be used to advantage to smooth the transition points – but it takes manual tweaking of the breakpoints and small adjustments in the series resistance to precisely fit the desired curve. It is an effort but the results are worth it – the desired curve is approximated with very good accuracy.

We will subtract 0.56 volts from each breakpoint to obtain the output voltages of the voltage divider. This gives us 1.69, 1.03, and 0.44 Volts. The source resistance of each voltage divider tap should be very small (no more than roughly 1 percent) compared to the series resistance in the diode network. Otherwise there will be error in the slopes – although this error can be compensated by reducing the slope resistor by the appropriate amount.

The easy way to design a multi-tap voltage divider is to establish the ratios of the resistors to the resistor whose low side is connected to ground. Then we will determine the smallest value of standard resistor to use to implement the voltage divider. The ratios of the resistors are just the ratios of voltage drops. So (beginning with R_A), the voltage drops are 0.44, $(1.03 - 0.44) = 0.59$, and $(1.69 - 1.03) = 0.66$. Normalizing each voltage drop by 0.44, the resistor ratios are: 1.00, 1.34, and 1.50. One candidate set of resistors is 51, 68, and 75 Ohms. There are other possibilities but this one will serve our example well. The source resistance at the R_A tap is $51 \parallel (68 + 75) = 37.6$ Ohms; the source resistance at the R_B tap is $(51 + 68) \parallel 75 = 46$ Ohms, and the source resistance at the top of R_C is that of the voltage source – which should be close to 0 Ohms.

Piecewise Linear Circuits

Use of piecewise linear circuits in sine wave generation

Although sine wave oscillators can be made to generate very pure sine waves, it is generally difficult to adjust the frequency since more than one component has to be changed simultaneously. A simpler method is to generate a triangle wave and then use a piecewise linear circuit to shape the wave into a sine wave approximation. Nearly all analog multifunction signal generators use this method (the other methods involve a direct non-linear approximation – is difficult to make work well but uses fewer parts). As a historical note, this method was developed at Stanford University in the late 1940's and acquired by the Hewlett-Packard Company which had a very close relationship with Stanford. The first commercial product to use this method was the HP202A which was introduced circa 1950.

Two piecewise linear circuits are used – one for positive polarity and one for negative polarity. Except for polarity both networks are identical. The use of six diode controlled breakpoints for each polarity produces a complete sine wave cycle approximation with 26 total segments. With optimum choice of the breakpoints it is possible to achieve a distortion level of less than 1 percent.

The student should observe the sine wave from an analog sine-triangle-square wave oscillator – such as an HP3312 or one of the old Wavetek models (not the Agilent synthesized generator used now) . It is virtually impossible to see the breakpoints – this is because the design capitalizes on the diode curve and also that the breakpoints are optimally chosen. The best way to observe the breakpoints is to use a notch filter at the fundamental frequency of the sine wave so that only the harmonics remain. The exact number of breakpoints used (nearly always 6 for each polarity) can then be clearly seen. The sine wave fundamental is gone but the distortion wave (i.e. the difference between the approximation and the true sine function) is clearly visible and is about a factor of 100 smaller than the sine wave.