

---

## DC to AC Converters

### DC to AC Converters:

There are several reasons why you might want to convert a DC signal to an AC signal:

- Transformers work at AC but not at DC. If you want to use a transformer to bump up a voltage, you need an AC signal.
- Brush-type DC Servo Motors need 3-phase AC. Brushless DC servo motors have three leads: phase A, B, and C. Personally, I take exception to using the term 'DC' and 'phase' in the same sentence, but that's just me.

A brushless DC servo motor (BLDC) is a 3-phase AC synchronous motor with permanent magnets. It's very much like a stepper motor (which has 2 phases), only this one has three. If you see a motor with permanent magnets and three leads, it's probably a BLDC motor.



Honda Electric Motorcycle. A brushless DC motor is located in the hubs.

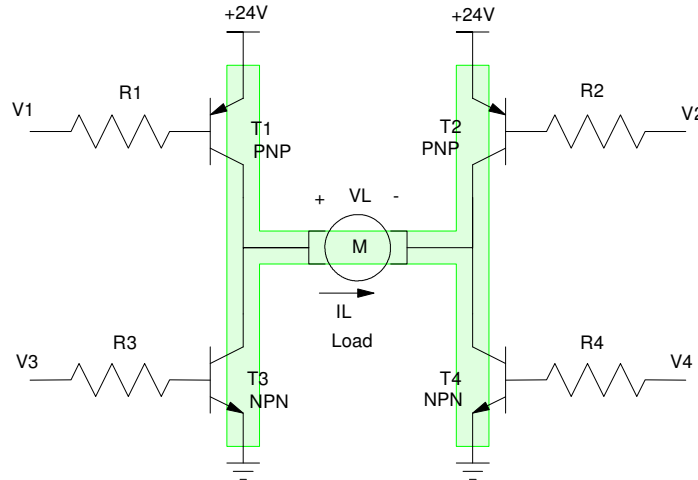
Brush-type DC motors (i.e. 2-lead) were used in the past due to the simplicity in powering them: connect a DC motor to a battery and it spins. Brushless DC motors (i.e. 3-lead) have significantly more power than a like brush-type DC motor and are more efficient. For electric bicycles, for example, the BLDC motors have 30% more range than the corresponding DC motor driven bicycle. They require electronics to drive them, however, which is why we're talking about them here.

Currently, BLDC motors are powerful enough to drive electric bicycles, with power ranging from 350 Watts to 1000 Watts and a cost of \$350 to \$1000. (You replace the front wheel of your bicycle with a wheel with a BLCD motor in the hub. With a larger BLDC motor, you can drive a motorcycle, such as the Honda Electric Motorcycle that's in development. With an even larger BLDC motor, you can drive a car.

Michelin is working on such a motor. Their bet is that shortly, the engine, transmission, gas tank, and starter will be replaced with the BLDC motor in the tire. At that point, a car manufacturer will simply attach one of Michelin's tires to their frame and call it a car.

### Single-Phase DC to AC Converter

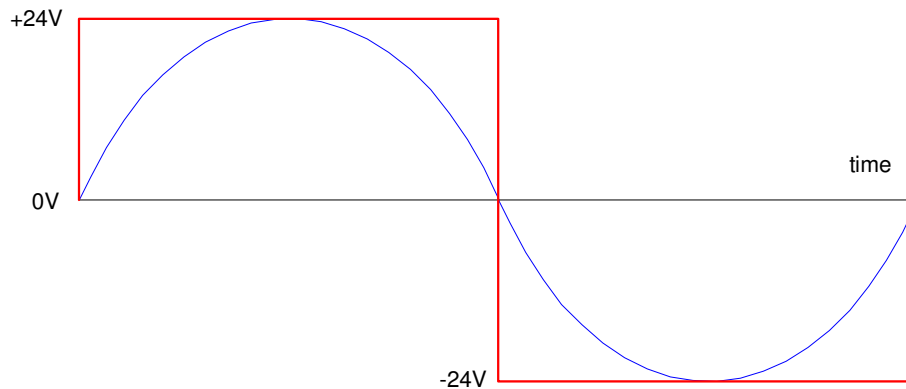
Suppose you want to drive a DC motor both forward and backwards. One way to do this is to use an H-bridge:



This allows you to control the direction of the motor by turning on and off different transistors:

| State     | T1  | T2  | T3  | T4  |
|-----------|-----|-----|-----|-----|
| VL = +24V | ON  | OFF | OFF | ON  |
| VL = -24V | OFF | ON  | ON  | OFF |
| VL = 0V   | OFF | OFF | OFF | OFF |

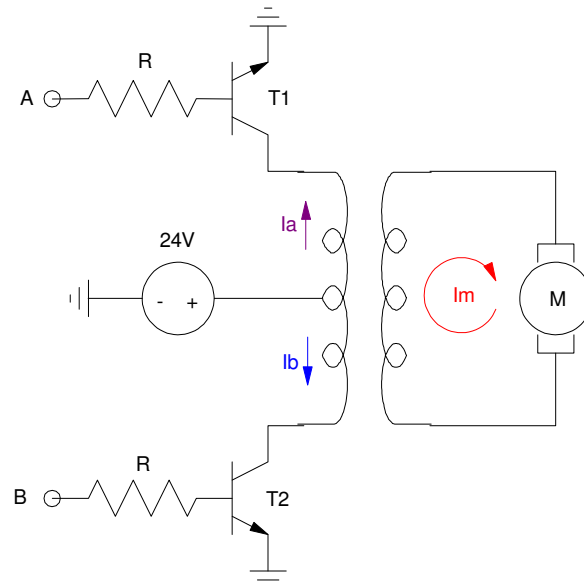
If you replace the motor with an AC motor (or a transformer), you can generate an +24V / -24V AC signal by cycling through these three states. If you switch between the first two, you are approximating a sine wave with a square wave:



Sine Wave (blue) and its Approximation (red)

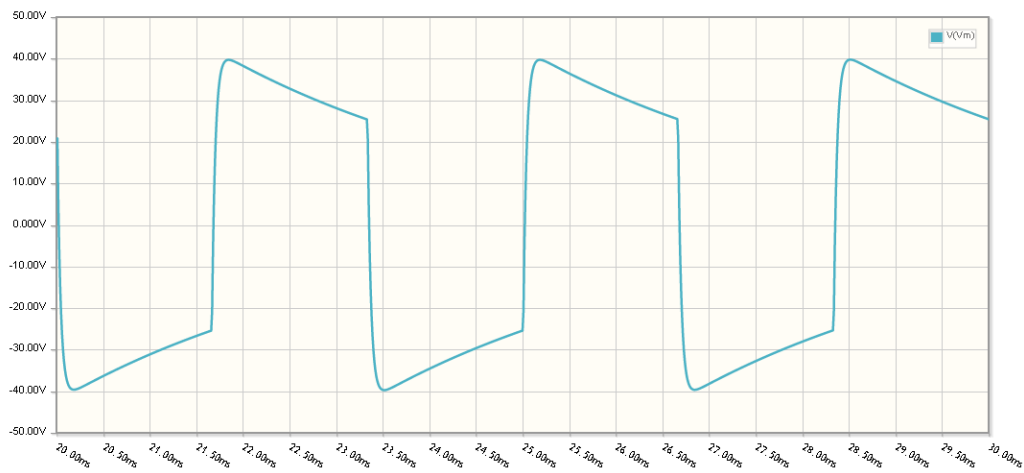
A second variation of this circuit uses a transformer with a center tap.

- When  $A = 5V$ , transistor T1 turns on and current flows up ( $I_a$ ). This induces clockwise current to the motor ( $I_m > 0$ ).
- When  $B = 5V$ , transistor T2 turns on and current flows down ( $I_b$ ). This induces counterclockwise current to the motor ( $I_m < 0$ ).



Another version of a DC to AC Converter.

For example, running this circuit in CircuitLab where A and B are 300Hz TTL (0V/5V) square waves 180 degrees out of phase produces the following voltage across the motor



AC voltage to the motor when T1 and T2 are alternately turned on at 300Hz

While these DC to AC converters may seem efficient, they're actually is only 81% efficient. To understand why, take the Fourier Transform of the voltage.

## Fourier Transform of a 50% Duty Cycle Square Wave

Assume  $f(t)$  is a square wave with a period of  $2\pi$  (i.e. a square wave approximating a sine wave from the previous H-bridge)

$$x(t) = \begin{cases} +24V & t < \pi \\ -24V & t > \pi \end{cases}$$

or in Matlab:

```
t = [0.001:0.001:1]' * 2 * pi;
x = 24*(t <= pi) - 24*(t > pi);
```

From before, the complex Fourier transform for  $x(t)$  is (all even terms are zero)

```
X1 = 2 * mean(x .* exp(-j*t))
X1 = -0.0960 -30.5576i

X3 = 2 * mean(x .* exp(-3*j*t))
X3 = -0.0960 -10.1856i

X5 = 2 * mean(x .* exp(-5*j*t))
X5 = -0.0960 - 6.1110i
```

| n  | 1       | 2 | 3       | 4 | 5      | 6 | 7      | 8 | 9      |
|----|---------|---|---------|---|--------|---|--------|---|--------|
| Xn | -j30.55 | 0 | -j10.18 | 0 | -j6.11 | 0 | -j4.36 | 0 | -j3.39 |

Fourier Coefficients for a 24V 50% Duty Cycle Square Wave.  
Note that  $\text{real}(X_n) = \text{cosine terms}$ ,  $-\text{imag}(X_n) = \text{sine terms}$

This means that

$$x(t) = \begin{cases} +24V & t < \pi \\ -24V & t > \pi \end{cases}$$

can also be expressed as

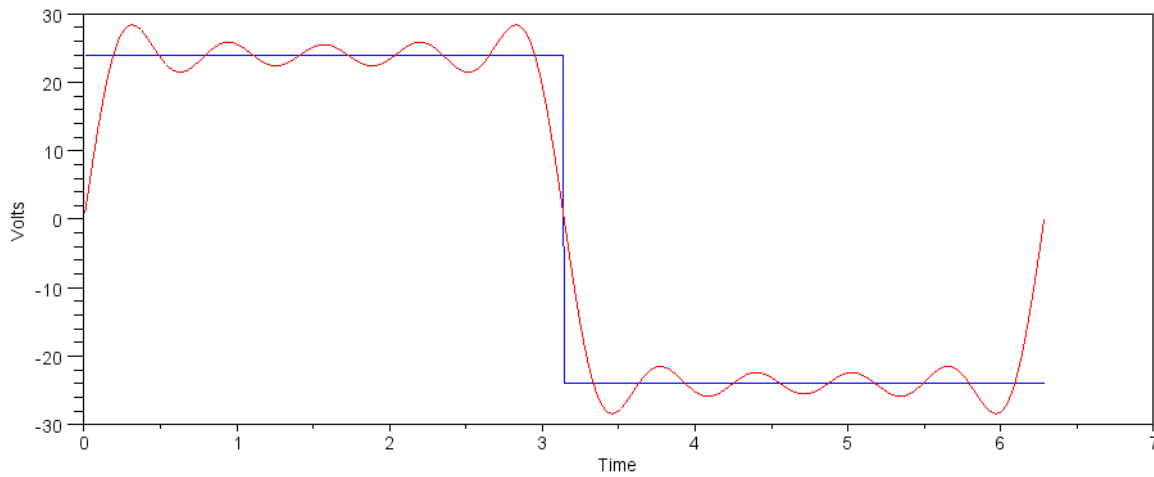
$$x(t) = 30.557 \sin(t) + 10.185 \sin(3t) + 6.111 \sin(5t) + 4.364 \sin(7t) + \dots$$

You can check your results by plotting the two:

```
xf = -imag(X1)*sin(t) -imag(X3)*sin(3*t) -imag(X5)*sin(5*t)
      -imag(X7)*sin(7*t) -imag(X9)*sin(9*t);
plot(t, x, t, xf)
xlabel('Time');
ylabel('Volts');
```

or if you prefer

```
xf = real( X1*exp(j*t) + X3*exp(j*3*t) + X5*exp(j*5*t) + X7*exp(j*7*t) );
plot(t, x, t, xf)
xlabel('Time');
ylabel('Volts');
```

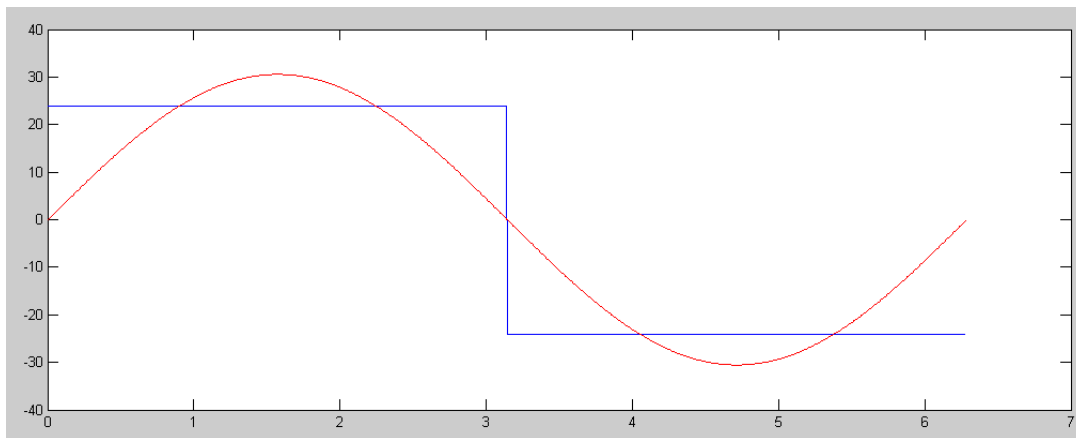


50% Duty Cycle Square Wave (blue) and its Fourier Series approximation taken out to the 7th harmonic (red)

**50% Duty Cycle DC to AC Converter (81% efficient):**

The entire signal,  $x(t)$  and it's 1st harmonic (  $x_f$  ) look like the following:

```
xf = real( X1 .* exp(j*t) );
plot(t, x, t, xf);
plot(t, x, 'b', t, xf, 'r');
```



Square Wave (blue) and its 1st Harmonic (red)

The power in each harmonic

$$E_n = \frac{1}{2}|X_n|^2$$

The percent of the total energy in the signal is

---

| n | Xn        | Power (Watts) | % Total |
|---|-----------|---------------|---------|
| 1 | -j30.5576 | 466.8896      | 81.06   |
| 2 | 0         | 0.0000        | 0       |
| 3 | -j10.1856 | 51.8780       | 9.01    |
| 4 | 0         | 0.0000        | 0       |
| 5 | -j6.1110  | 18.6771       | 3.24    |
| 6 | 0         | 0.0000        | 0       |
| 7 | -j4.3647  | 9.5299        | 1.65    |
| 8 | 0         | 0.0000        | 0       |

The percent of the energy in the 1st harmonic is 81%:

```
>> Pout = mean(xf .^ 2)
Pout = 466.8896
>> Pin = mean(x .^ 2)
Pin = 576
>> eff = Pout / Pin
eff = 0.8106
```

or equivalently:

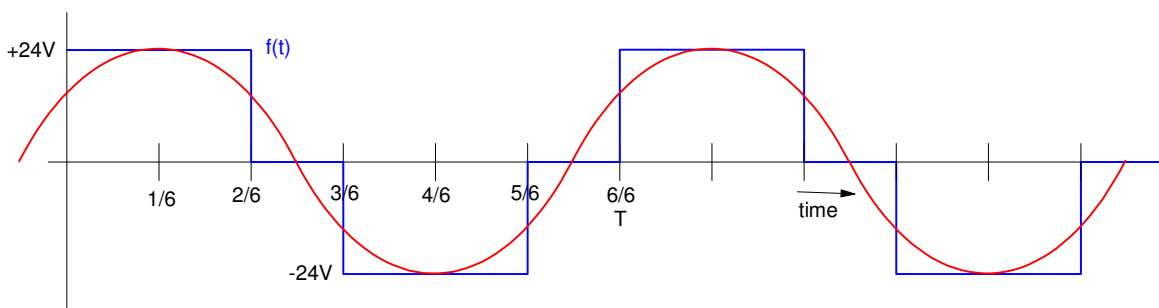
```
>> eff = 0.5*abs(X1)^2 / mean(x .^ 2)
eff = 0.8106
```

### Improved DC to AC Converter (91% efficient)

With an H-bridge, you don't have to connect to +24V or -24V. You have a 3rd option: turn off the H-bridge and let the voltage be zero.

A slightly better approximation to a sine wave uses three states with the output being

- +24V for 1/3rd of the time,
- -24V for 1/3rd of the time, and
- 0V for 1/6th of the time when you go -24V as well as when you go to +24V:



To find the efficiency of this DC to AC converter, input this signal into Matlab:

Start with t going from 0 to 1 seconds to make the regions easier:

```
t = [0:0.001:1]';
x = 24*(t<2/6) - 24*(t>3/6) .* (t<5/6);
```

Now rescale t so that its period is  $2\pi$

```
t = t * 2 * pi;
```

Compute the Fourier coefficients. All we really care about is the 1st harmonic:

```
X1 = 2 * mean(x .* exp(-j*t))
X1 = 13.2107 -22.9092i
```

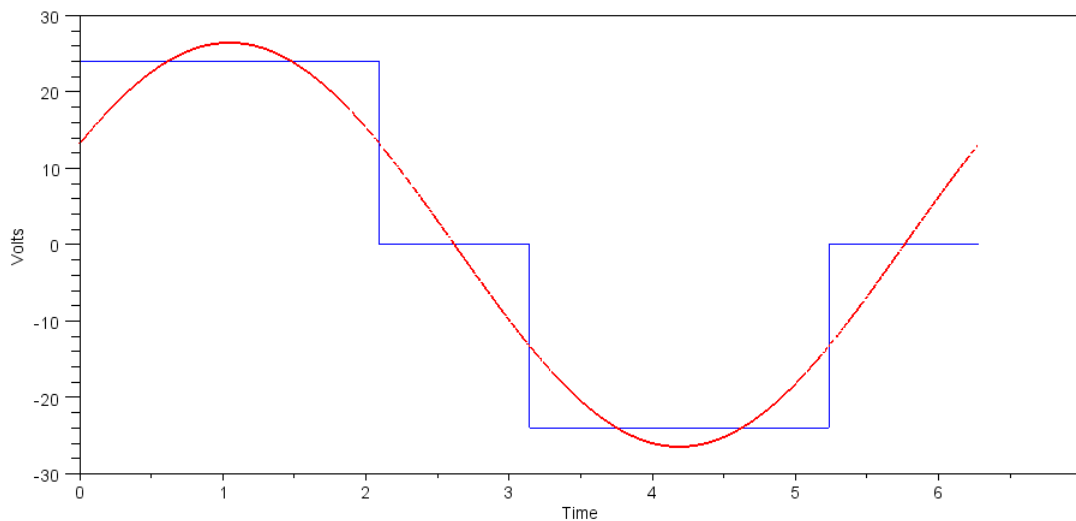
The efficiency is then

```
>> eff = 0.5*abs(X1)^2 / mean(x.^2)
eff = 0.9111
```

91.11% of the energy is in the 1st harmonic. This DC to AC converter is likewise 91.11% efficient.

Just for fun, plot the 1st-harmonics of x(t) vs. the original signal:

```
xf = real( X1 .* exp(j*t) );
plot(t,x,'b',t,xf,'r');
```



f(t) and its First Harmonic

The energy in each of the 1st eight harmonics is:

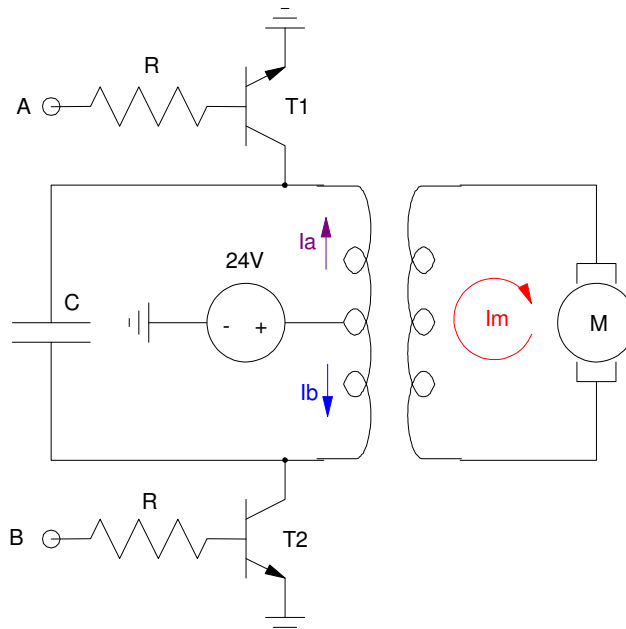
| n | $X_n$             | Power (Watts) | % Total |
|---|-------------------|---------------|---------|
| 1 | 13.2107 -22.9092i | 349.6774      | 91.11   |
| 2 | 0                 | 0             | 0       |
| 3 | 0                 | 0             | 0       |
| 4 | 0                 | 0             | 0       |
| 5 | -2.6518 - 4.5650i | 13.9358       | 3.63    |
| 6 | 0                 | 0             | 0       |
| 7 | 1.8805 - 3.2843i  | 7.1615        | 1.87    |
| 8 | 0                 | 0             | 0       |



**Variation 3: 98.79% Efficient**

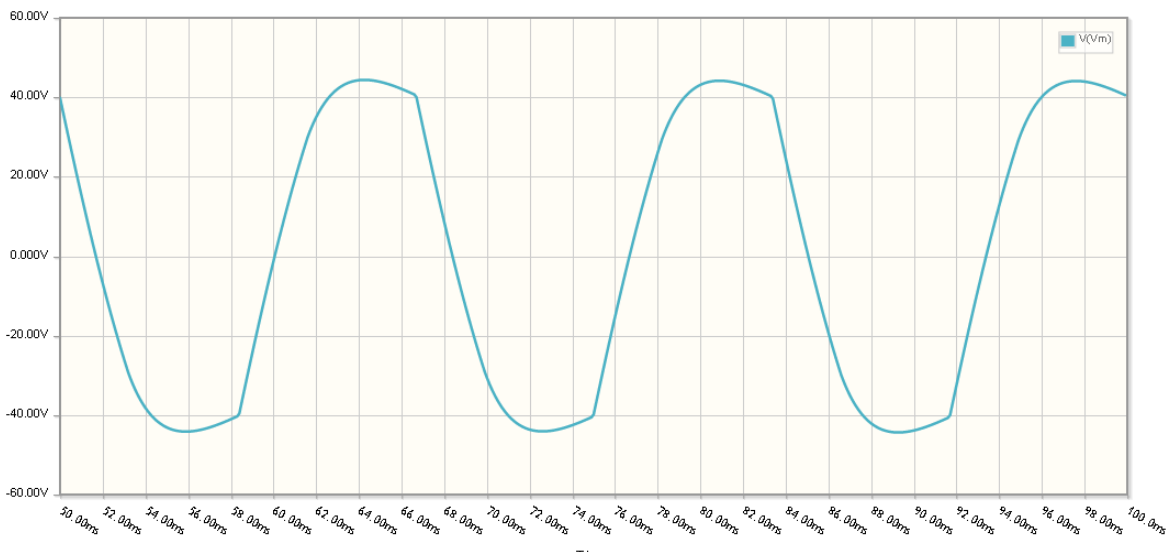
If you add a capacitor across the primary side of the transformer

- The capacitor helps to remove the delta functions in voltage as Ia switches to Ib
- It also adds filtering - removing some of the high-frequency terms on the primary side



Variation of DC to AC converter  
A capacitor is added to provide filtering

If you set  $C = 200\mu\text{F}$  (by trial and error) you get the following waveform across the motor (modeled as a 100 Ohm resistor)



Vm: PartSim simulation of DC to AC converter running at 60hz with  $C = 200\mu\text{F}$

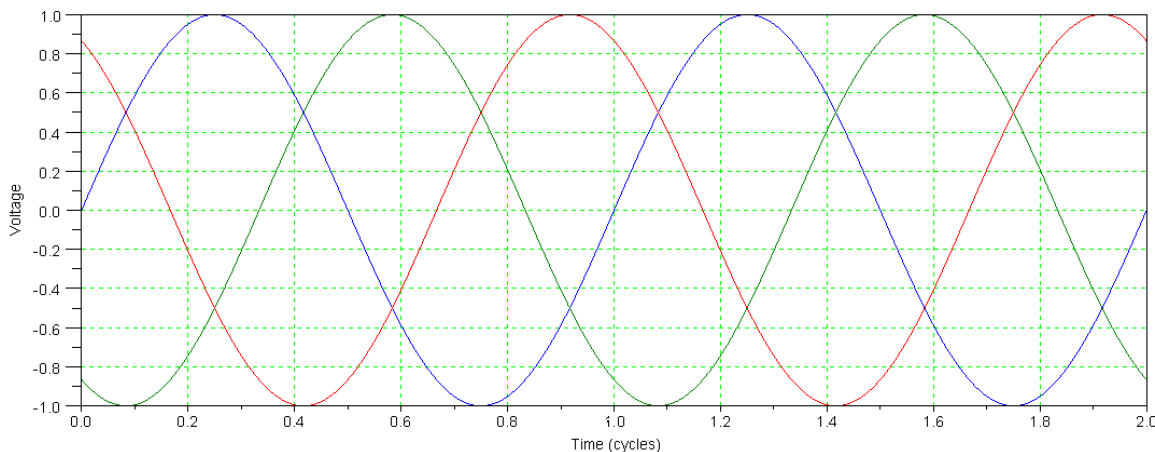
The frequency content of the waveform to the motor is then

| n | Xn                 | Power (Watts) | % Total |
|---|--------------------|---------------|---------|
| 1 | -30.2949 -35.1378i | 1,076.2       | 98.79%  |
| 2 | 0                  | 0             | 0       |
| 3 | -4.6643 + 1.6672i  | 12.27         | 1.13%   |
| 4 | 0                  | 0             | 0       |
| 5 | -1.1407 + 0.5278i  | 0.79          | 0.07%   |
| 6 | 0                  | 0             | 0       |
| 7 | -0.5034 + 0.2105i  | 0.15          | 0.01%   |
| 8 | 0                  | 0             | 0       |

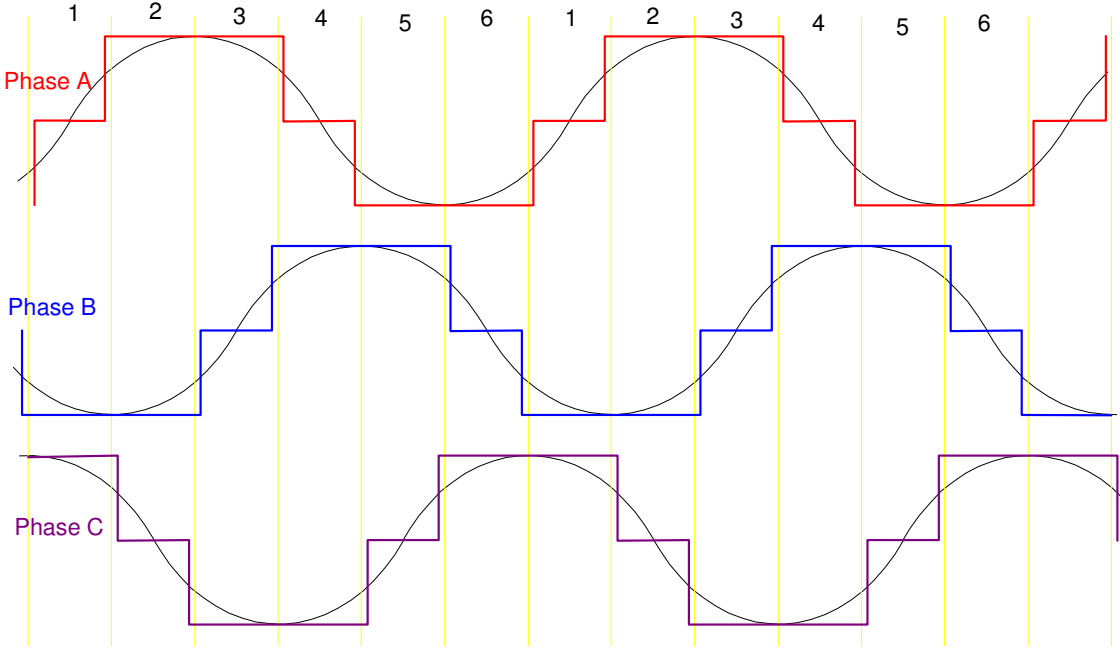
### 3-Phase DC to AC Conversion: BLDC Motor Controller

A BLDC motor is short for "Brushless DC" motor. I personally dislike using the word "phase" and "DC" in the same sentence. It's actually a 3-phase AC synchronous motor. You'll have to ask someone else why it isn't called what it is...

To generate a 3-phase AC signal, you generate three separate AC signals, each 120 degrees out of phase:



The corresponding digital signals for phase A, B, and C are then as follows:



Approximation for a 3-Phase AC Signal Using Six Steps