3 Phase Inverter MATLAB Simulation

2019142208 Joshua

Introduction

In this MATLAB Simulation, we simulate a three-phase inverter using the mathematical model discussed during class. In this report, we show that the three-phase inverter can be mathematically modelled wherein the line-to-line voltages can be represented by the switching function discussed in class. We also discuss the effects of overmodulation and how the third harmonic injection can be used to keep the inverter well within linear mode. We also observe the effects of variable voltage, variable frequency, as well as a quick look on the Fourier Transform to check for harmonics.

Code

We start by asking the user for simulation options: inject 3rd harmonic component, do a frequency sweep, and/or do amplitude sweep for the input voltage. Setting the 3rd injection off, causes the 3rd harmonic to be multiplied to 0 (line 88) else will be multiplied to 1. Setting frequency sweep causes the pole voltages to have increasing frequency as the simulation runs. Similar setting applies to the amplitude sweep.

```
5   %% Ask User for Simulation Options
6
7   prompt_3rd = 'Inject 3rd Harmonic Component? [Yes: 1, No: 0]: ';
8   cond_3rd = input(prompt_3rd,"s");
9   if cond_3rd == "1"
10      third_harmonic_injection = 1;
11  else
12      third_harmonic_injection = 0;
13  end
14
15  prompt_f = 'Frequency Sweep? [Yes: 1, No: 0]: ';
16  cond_f = input(prompt_f, "s");
17  if cond_f == "1"
18      frequency_sweep = 1;
19  else
20      frequency_sweep = 0;
21  end
22
23  prompt_a ='Amplitude Sweep? [Yes: 1, No: 0]: ';
24  cond_a = input(prompt_a,"s");
25  if cond_a == "1"
26      amplitude_sweep = 1;
27      amp_init = 220; % starting amplitude of sweep (RMS)
28      amp_fin = 720; % final amplitude of sweep (RMS)
29  else
30      amplitude_sweep = 0;
31  end
32
```

<Figure 1: Simulation Options>

The following section finishes the setup process by declaring the DC bus voltage, frequencies, carrier, and reference waveforms.

```
%% Simulation Setup
% DC comes from a three-phase DBR connected to 3 ph Voltage Source;
% 220 Vrms
Vdc = sqrt(2)*sqrt(3)*220;
Vdc_2= Vdc/2;

% variable voltage and variable frequency....
% Frequency command
f=60;
% Carrier frequency
fs = 3e3;
% This is for plotting.
Fs = 1e6;
dt = 1/Fs;

% Number of cycles and time duration
if frequency_sweep== 0
    no=4;
else
    no = 6;
end

T = no*1/f;
t = 0:dt:T-dt;

Vref = 380;


if amplitude_sweep == 0
    Vm = Vref/sqrt(3)*sqrt(2);
else
    sweep_start_amp = amp_init/sqrt(3)*sqrt(2);
    sweep_end_amp = amp_fin/sqrt(3)*sqrt(2);
    Vm = linspace(sweep_start_amp,sweep_end_amp,length(t));
end

% To control your voltage output
ma = Vm/Vdc_2;

% Phase voltage magnitude(Pole Voltage)
if frequency_sweep == 0
   Van= Vm.*sin(2*pi*f*t);
   Vbn= Vm.*sin(2*pi*f*t-120/360*2*pi);
   Vcn= Vm.*sin(2*pi*f*t+120/360*2*pi);
else
   Van= Vm.*chirp(t,60,0.03,120,"linear",-90);
   Vbn= Vm.*chirp(t,60,0.03,120,"linear",-90-120);
   Vcn= Vm.*chirp(t,60,0.03,120,"linear",-90+120);
end

% To inject a 3rd harmonic component
V_3rd = -Vm/6.*sin(2*pi*f*3*t);
V_3rd = third_harmonic_injection*V_3rd;

%line to line voltage
Vab = Van-Vbn;
Vbc = Vbn-Vcn;
Vca = Vcn-Van;

% triangular waveform (carrier waveform)
Vtri = sawtooth(2*pi*fs*t,1/2)*Vdc_2;

% Reference Voltages used in the modulation
Va_cmd = Van-V_3rd;
Vb_cmd = Vbn-V_3rd;
Vc_cmd = Vcn-V_3rd;
```

<Figure 2: Simulation Setup>

For the modulation, the switching action is similar with the single-phase inverter: when the reference signal's magnitude is higher than the carrier, the switch on that pole is on (S =1) else, is off (S = 0).

```
for i = 1:length(t)
    if Va_cmd(i) >= Vtri(i)
        Sa = 1; % switch is on, same goes for other phases
    else
        Sa = 0; % switch is off, same goes for other phases
    end

    if Vb_cmd(i) >= Vtri(i)
        Sb = 1;
    else
        Sb = 0;
    end

    if Vc_cmd(i) >= Vtri(i)
        Sc = 1;
    else
        Sc = 0;
    end


    Van_pwm(i) = Vdc/2*(2*Sa-1);
    Vbn_pwm(i) = Vdc/2*(2*Sb-1);
    Vcn_pwm(i) = Vdc/2*(2*Sc-1);

    % phase voltages can be represented in terms of switching functions
    % and pole voltage
    Vas(i) = Vdc/3*(2*Sa-Sb-Sc);
    Vbs(i) = Vdc/3*(2*Sb-Sc-Sa);
    Vcs(i) = Vdc/3*(2*Sc-Sa-Sb);

    % Line to line voltage (Vab = Vas- Vbs = Van - Vbn)
    Vab_pwm(i) = Vas(i) - Vbs(i);
    Vbc_pwm(i) = Vbs(i) - Vcs(i);
    Vca_pwm(i) = Vcs(i) - Vas(i);

```

<Figure 3: Modulation>

Since the output requires a low pass filter to extract the fundamental, we also add the filter in this for loop to avoid adding another loop at the end.

```

    % With the output, we take extract the fundamental using a low
    % pass filter. We use the indicated filter from above.

    % low pass filter
    if i < 3
        if i == 1
            Vabf(i) = Vab_pwm(i);
            Vbcf(i) = Vbc_pwm(i);
            Vcaf(i) = Vca_pwm(i);
        else
            Vabf(2) = Vab_pwm(1);
            Vbcf(2) = Vbc_pwm(1);
            Vcaf(2) = Vca_pwm(1);
        end
    else
        % IIR Filter
        Vabf(i)= 0.999*Vabf(i-1)+0.001*Vab_pwm(i);
        Vbcf(i)= 0.999*Vbcf(i-1)+0.001*Vbc_pwm(i);
        Vcaf(i)= 0.999*Vcaf(i-1)+0.001*Vca_pwm(i);
    end
end

```

<Figure 4: Low Pass Filter for the Line-to-Line Voltages>

Once the modulation and filter for loop is done, we plot the results.

```
%% Plot Modulation Results

figure; sgtitle("Brief Summary of Modulation Results");
subplot(3,1,1); plot(t,Van_pwm); grid on;
xlabel("time (s)"); ylabel("Van");
subplot(3,1,2); plot(t,Vas); grid on;
xlabel('time (s)'); ylabel("Vas");
subplot(3,1,3); plot(t,Vab_pwm); grid on;
xlabel("time (s)"); ylabel("Vab");

% Pole Voltages
figure; sgtitle("Full Modulation Results");
subplot(3,3,1); plot(t,Van_pwm); grid on;
xlabel("time (s)"); ylabel("Van");
subplot(3,3,2); plot(t,Vbn_pwm); grid on;
xlabel("time (s)"); ylabel("Vbn");
subplot(3,3,3); plot(t,Vcn_pwm); grid on;
xlabel("time (s)"); ylabel("Vcn");


% Line to Line PWM

subplot(3,3,7); plot(t,Vab_pwm); grid on;
xlabel("time (s)"); ylabel("Vab");
subplot(3,3,8); plot(t,Vbc_pwm); grid on;
xlabel("time (s)"); ylabel("Vbc");
subplot(3,3,9); plot(t,Vca_pwm); grid on;
xlabel("time (s)"); ylabel("Vca");

% Phase Voltages

subplot(3,3,4); plot(t,Vas); grid on;
xlabel("time (s)"); ylabel("Vas");
subplot(3,3,5); plot(t,Vbs); grid on;
xlabel("time (s)"); ylabel("Vbs");
subplot(3,3,6); plot(t,Vcs); grid on;
xlabel("time (s)"); ylabel("Vcs");
```

<Figure 5: Modulation Results>

```
%% Plot PWM Result
figure; sgtitle(" Brief PWM Result");
plot(t, Vab_pwm, t, Vabf); grid on;
xlabel("time(s)");
ylabel("Vab and Vab\_LPF");

figure; sgtitle("PWM Results");
subplot(3,1,1); plot(t, Vab_pwm, t, Vabf); grid on;
xlabel("time(s)"); ylabel("Vab and Vab\_LPF");
subplot(3,1,2); plot(t, Vbc_pwm, t, Vbcf); grid on;
xlabel("time(s)"); ylabel("Vbc and Vbc\_LPF");
subplot(3,1,3); plot(t, Vca_pwm, t, Vcaf); grid on;
xlabel("time(s)"); ylabel("Vca and Vca\_LPF");

figure; sgtitle("Output of 3\phi-phase Inverter");
plot(t,Vabf,t,Vbcf,t,Vcaf); grid on;
xlabel("time(s)"); ylabel("Magnitude");
```

<Figure 6: PWM Result>

Finally, we setup the code for the harmonics using fft( ).

```
%% Check Harmonics
x = Vab_pwm; % check harmonics on Vab line to line voltage
% x = Van_pwm;

f = 60;
Fs = 1e6;

L = length(t);
T = 1/Fs;
t = (0:L-1)*T;

Y = fft(x);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);

figure;
f = Fs*(0:(L/2))/L;
plot(f,P1); grid minor;
title("Single-Sided Amplitude Spectrum of Vab\_pwm");
```

<Figure 7: Harmonics>

Results

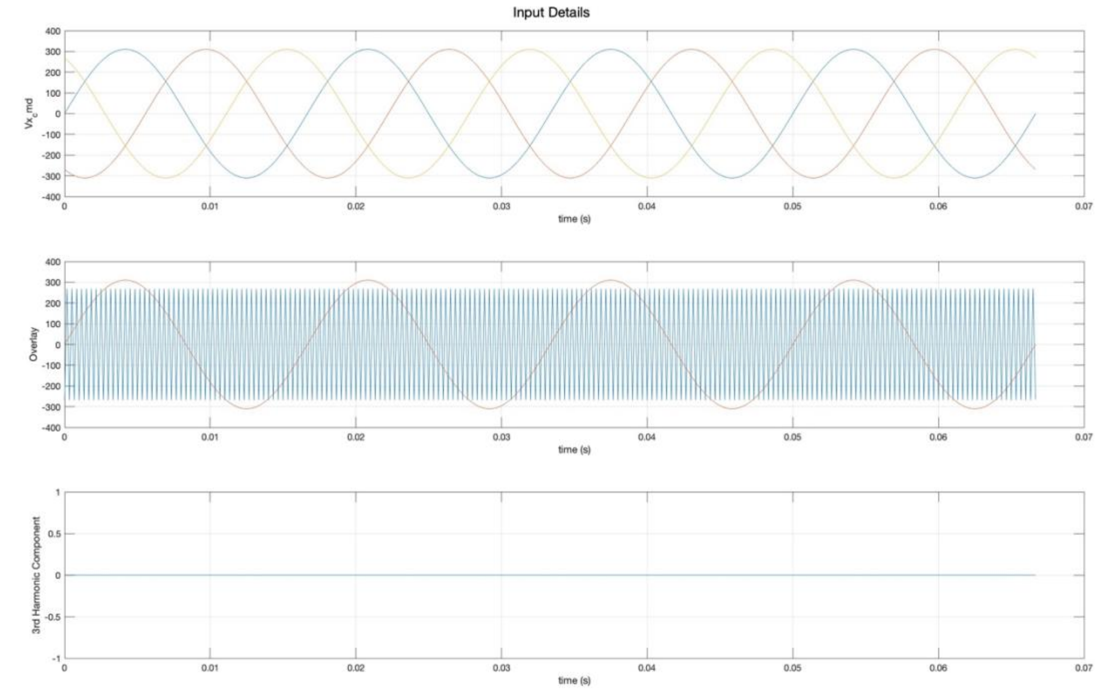
We start with the basic case: no harmonic injection, no sweeps, with desired output of 380 V RMS (line-to-line).

```
Command Window
Inject 3rd Harmonic Component? [Yes: 1, No: 0]: 0
Frequency Sweep? [Yes: 1, No: 0]: 0
Amplitude Sweep? [Yes: 1, No: 0]: 0
fx >>
```

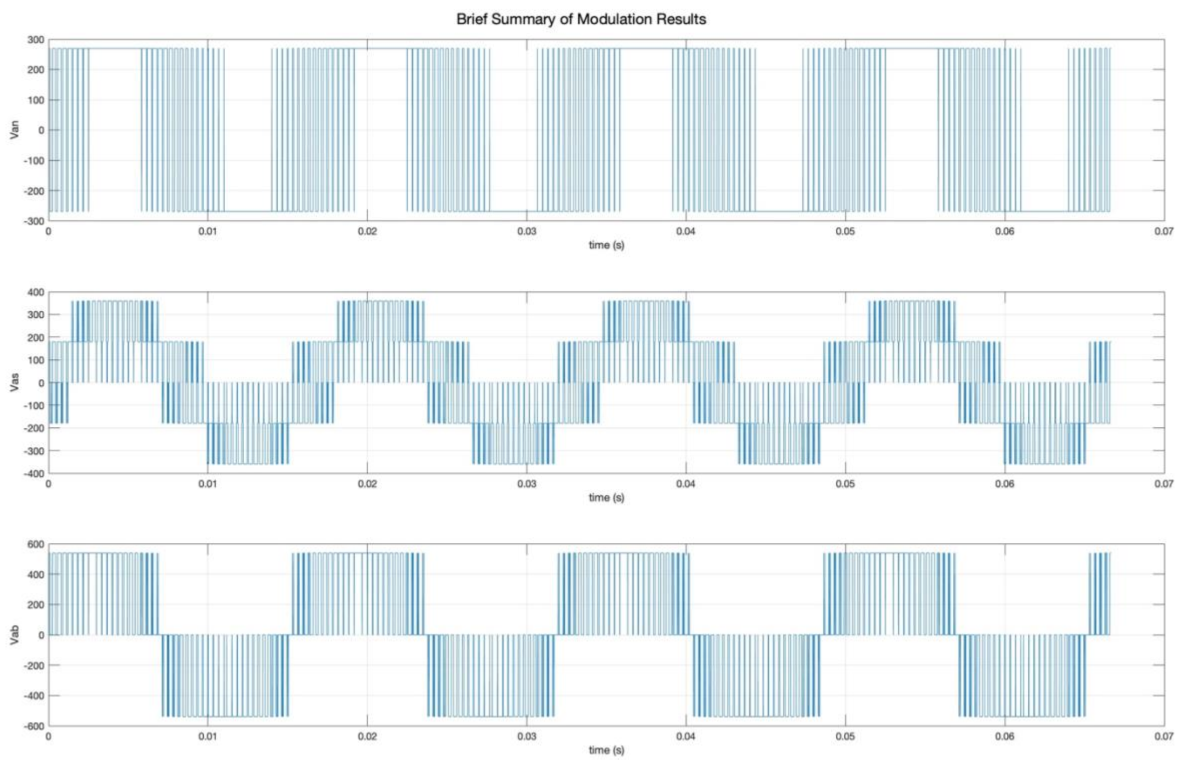
<Figure 8: Command Window for base case>

As shown in figure 9 (next page), we see that the voltage that we want at the output exceeds the magnitude of the carrier waveform, which is the DC bus limit for the inverter. We also see that there is no 3rd harmonic component in the waveform.

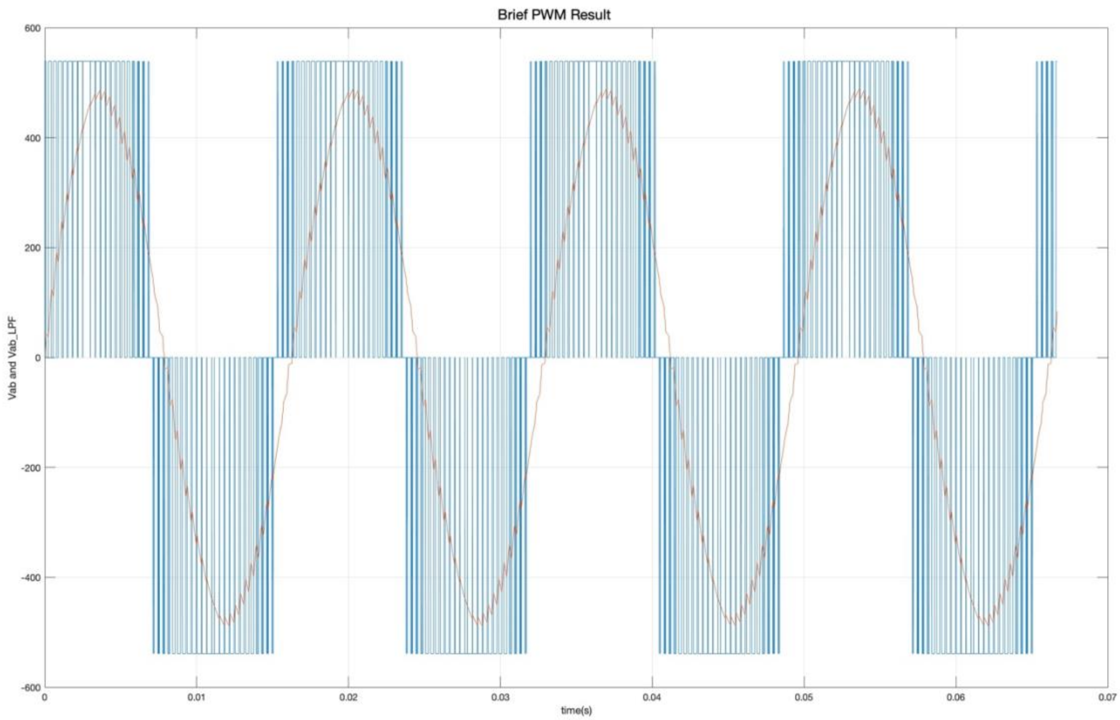
Figure 10 (next page) shows the results of the modulation. As seen in the first graph, the pole voltage with respect to the fictitious node has a segment of time where the switch is kept on (because of overmodulation). The second graph show that the phase voltage indeed has 5 levels as discussed, and the final graph shows the unipolar SPWM resulting waveform.



<Figure 9: Input Details>

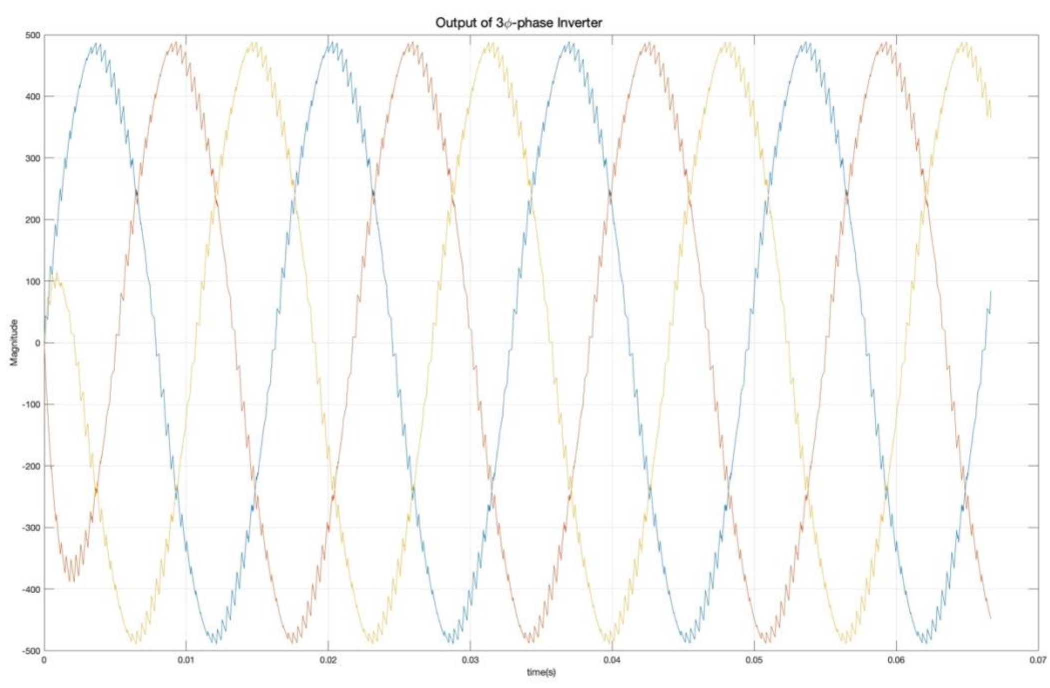


<Figure 10: Modulation Results>



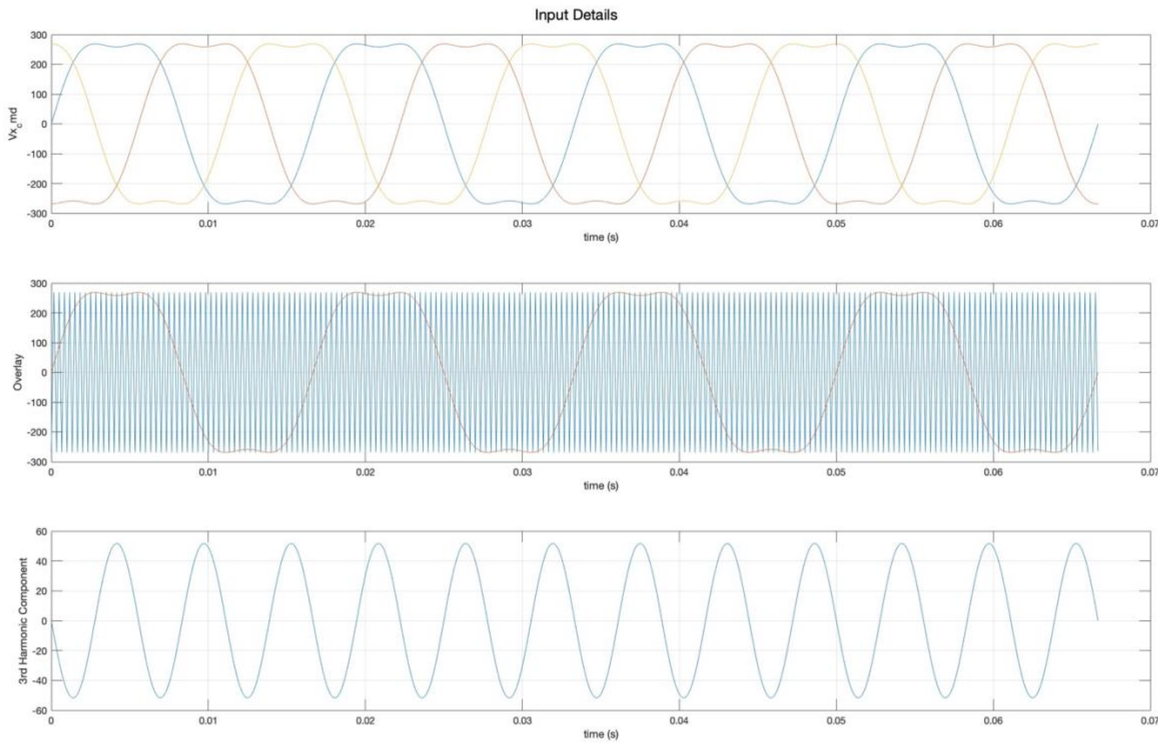
<Figure 11: PWM resulting waveform + filtered result>

Shown in figure 11 is the filtered PWM waveform to extract the fundamental. We can also see that there is delay due to the filter. Figure 12 below shows the overall output of the three-phase inverter.



<Figure 12: Output of the 3-phase SPWM Inverter>

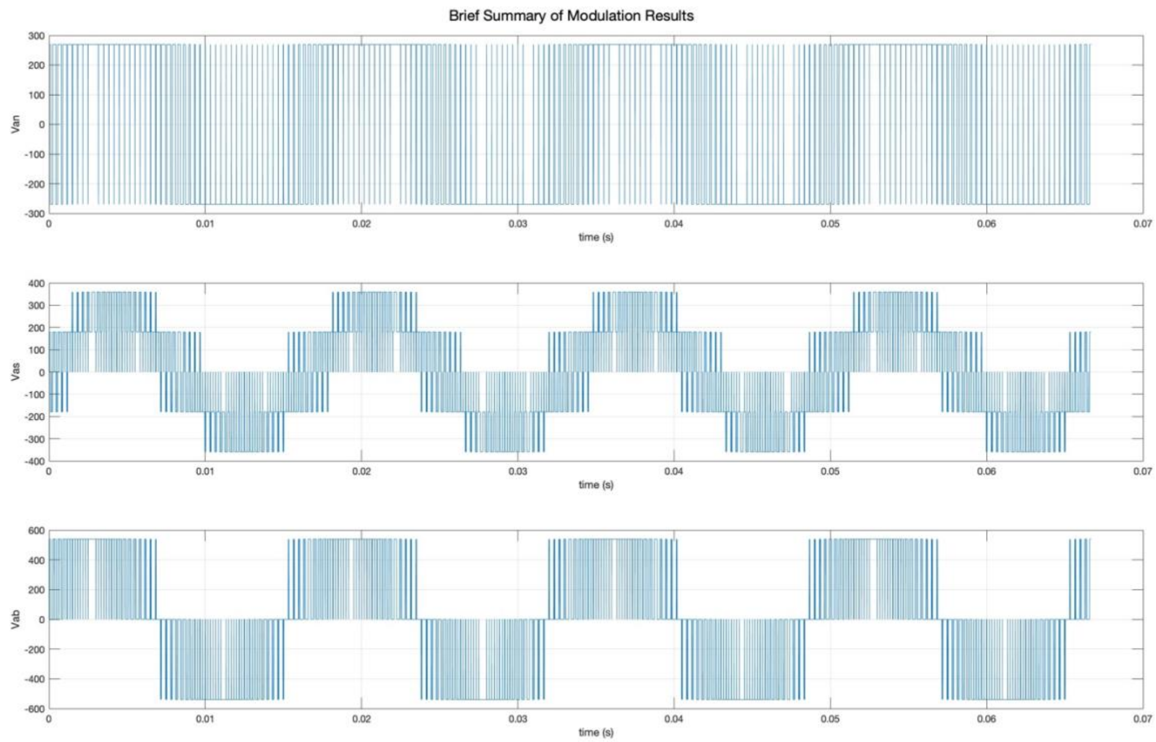
Before we check the harmonics of the line-to-line voltage, we first observe the results of the base case with the 3rd harmonic injection option selected. Figure 13 shows the input waveform. As seen, in the second graph, the reference voltage has the 'tip' cut-off to ensure that the inverter avoids overmodulation. The third graph shows the third harmonic component injected.



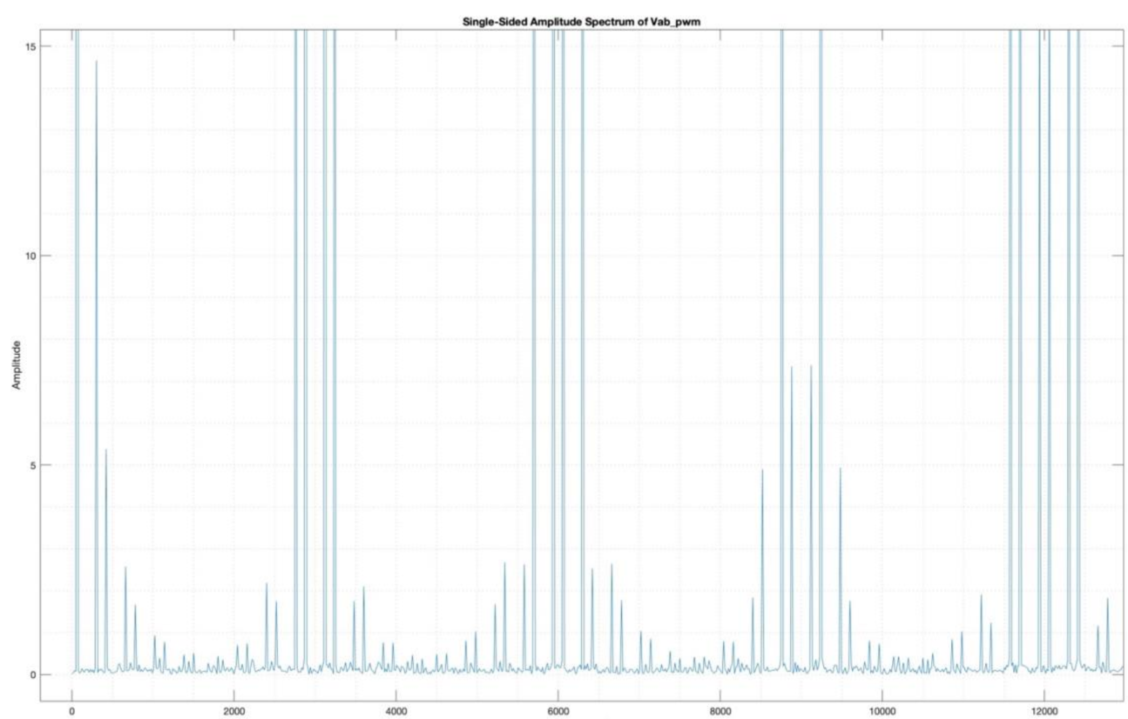
<Figure 13: 3rd Harmonic Injection Input>

The use and benefit of injecting 3rd harmonic component waveform to the input of the inverter is shown in figure 14 (next page). Clearly from the first graph, the white space has been removed if not reduced which is an indication of improvement in the input and modulation of the inverter. Figure 15 (next page) shows that the PWM output follows the same form - the extracted fundamental is still well within the DC bus limit.

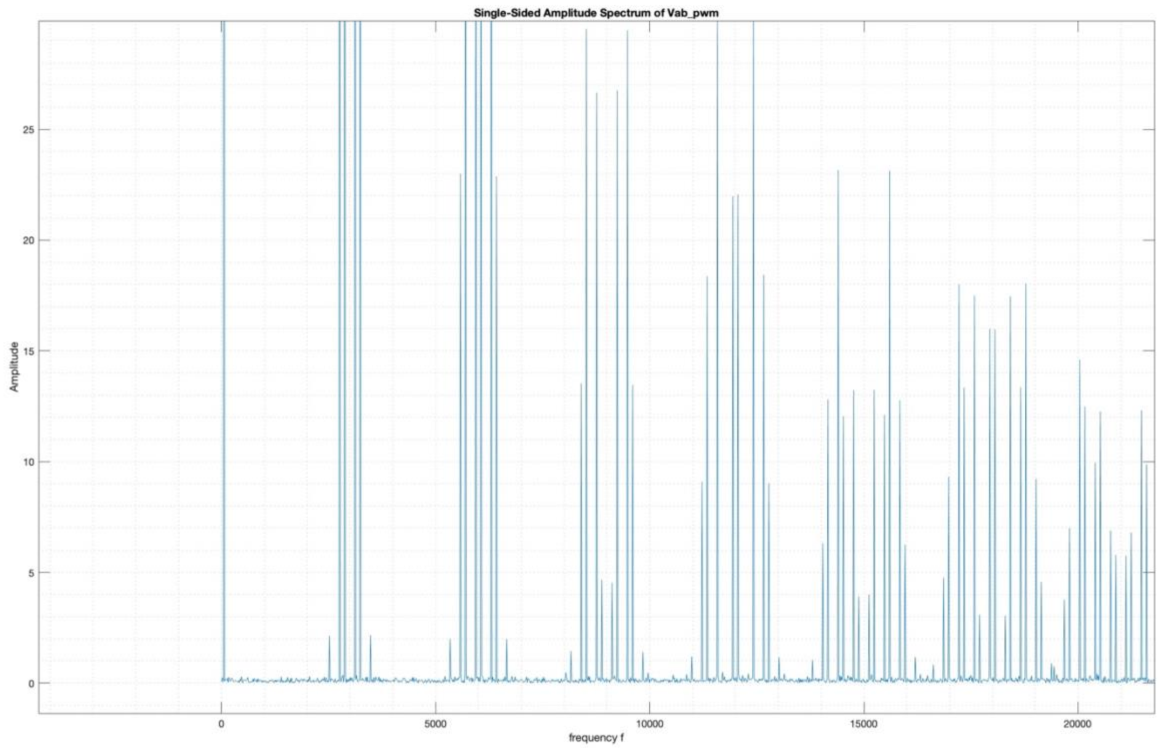
We compare the harmonics of both cases. Figure 16 and 17 shows the harmonics for the base case and 3rd harmonic injection case respectively. As observed, the overmodulated base case suffers from harmonics around the fundamental while injecting the 3rd harmonic component removes the harmonics around the fundamental. This shows that the 3rd harmonic injection technique allows the inverter not to be overmodulated.



<Figure 14: Modulation Results with 3rd Harmonic Injection>



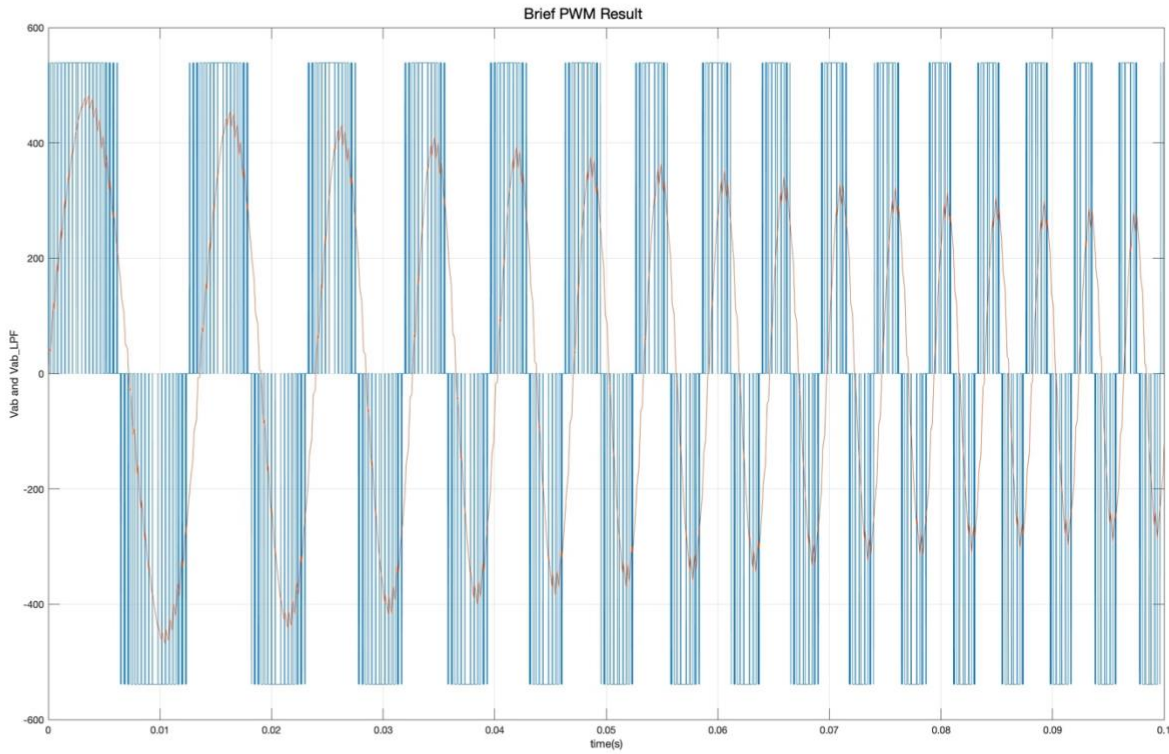
<Figure 15: Harmonics of the Base Case>



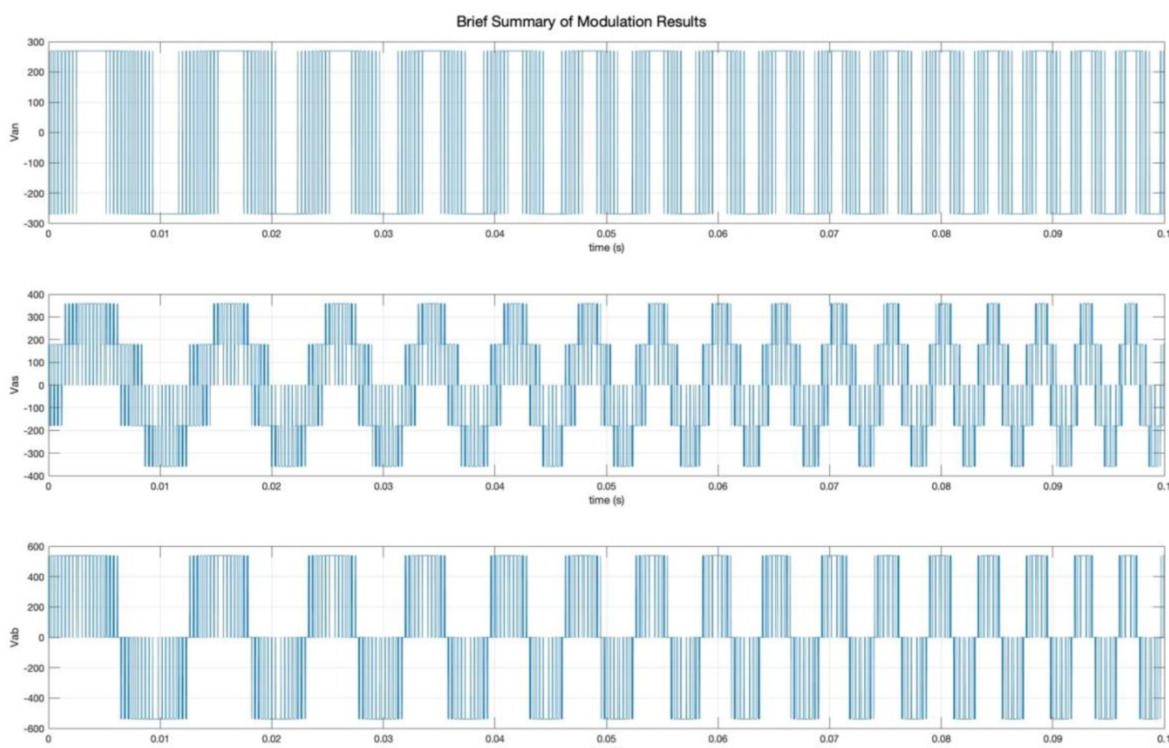
<Figure 16: Harmonics with 3rd Harmonic Injection>

We now observe the effects of varying frequency using the frequency sweep option. Our focus would be on the filtered output and its respective PWM waveform shown in figures 17 and 18. Note that the frequency sweep is set starting from 60 Hz to 120 Hz.

Figure 17 shows that increasing frequency causes the filtered fundamental to have its magnitude lower due to the low pass filter. By figure 18, we can verify that overmodulation does not depend on the frequency as we can still see that white space in the middle of each section of each set of pulses.

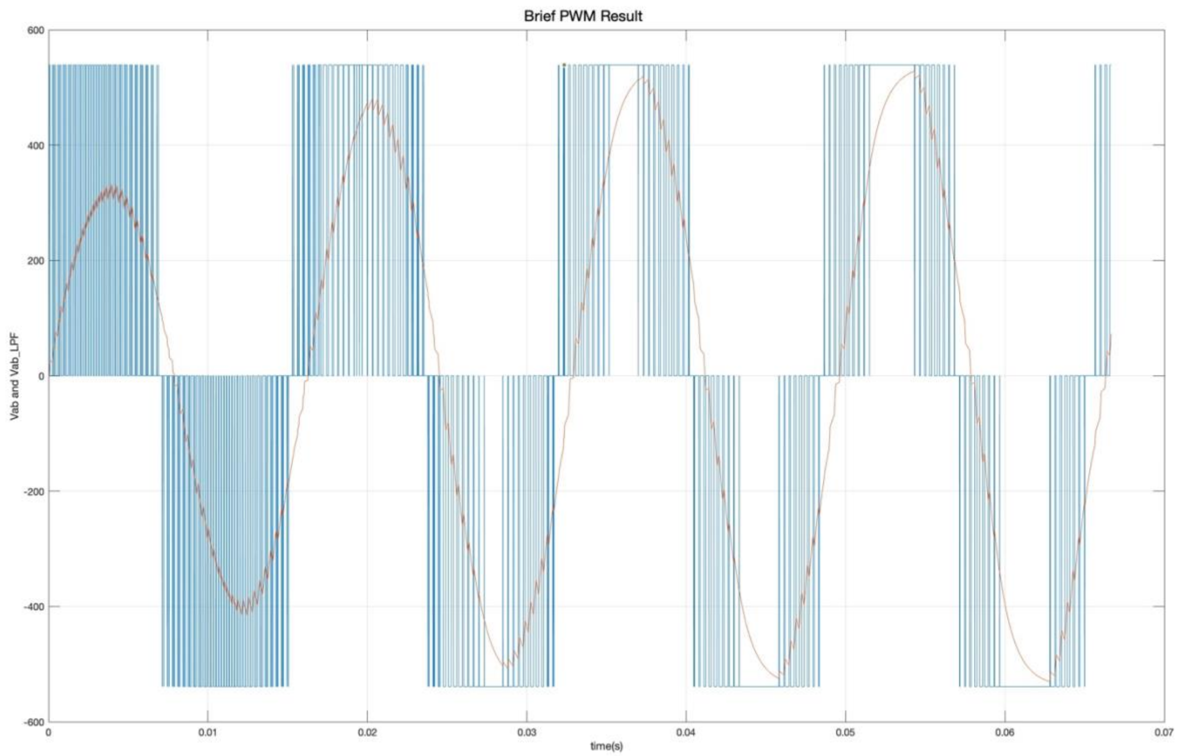


<Figure 17: PWM Result for Frequency Sweep>

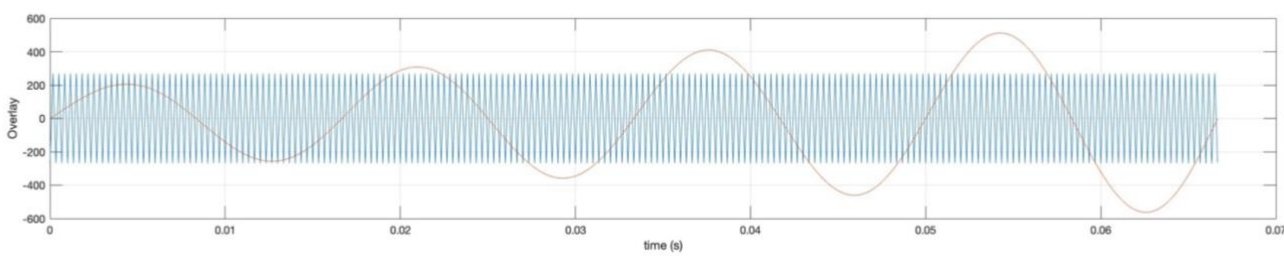


<Figure 18: Frequency Sweep Modulation Results>

Finally, we observe the effects of varying the magnitude using the amplitude sweep option. In this case, the voltage sweeps from 220 V RMS to 720 V RMS. We focus on figure 19 which shows the PWM result with the filtered sine wave on top of the graph. As shown below, increasing the voltage amplitude of the reference causes the inverter to be overmodulated as indicated by the increasing white space in between each set of pulses. To add to this observation, figure 20 shows that the overmodulation is caused by the fact that the increasing reference voltage would cause the reference voltage to be bigger than the voltage at the DC bus, hence the white spaces.



<Figure 19: PWM Output + Fundamental>



<Figure 20: Increasing Voltage causes Overmodulation>