Management Center Innsbruck

Department of Technology & Life Sciences

Master's program Mechatronics & Smart Technologies

Laboratory report

composed as part of the course
Drive Systems Laboratory (MECH-M-2-ATS-ATS-LB)

about

Drive Systems and different control strategies for a PMSM/BLDC motor

from

Paul Obernesser, Jakob Spindler

Study program	Master's program Mechatronics & Smart Technologies
Year	MA-MECH-23-VZ
Course	Drive Systems Laboratory (MECH-M-2-ATS-ATS-LB)
Name of lecturer	Bernhard Hollaus and Daniel McGuiness
Submission deadline	May 24, 2024

May 24, 2024

Contents

Chapter 1

Overwiev of the lab

The purpose of this lab is to compare different control strategies for a permanent magnet synchronous motor (PMSM), these are Sensored 120 Degrees block commutation with PWM, Sensorless 120 Degrees block commutation with PWM and Sensorless field oriented control (FOC). In the lab the control strategies are studied in detail and the performance of the control strategies is compared. Also the the influence of the speed and applied loads is studied.

The control strategies are implemented on the PSoC 4 Motor Control Evaluation Kit in combination with the PSoC 4 Pioneer Kit. A meassurement board created by the MCI is used to measure the phase currents and voltages of the motor.

Chapter 2

Sensored BLDC 120 Block Commutation

This control strategy is based on three hall sensors with 120° electrical separation. The hall sensors are used to determine the rotor position and therefore the commutation sequence. The commutation sequence is shown in Figure 2.1.

2.1 Commutation Sequence

The commutation stratedy implemented in this laboratory is based on the 120° block commutation. Only the high side switches are chopped with the PWM signal. The low side switches are not chopped and are on for the whole 120° block. The commutation sequence is shown in Figure 2.1.

This commutation sequence is implemented usindg a lookup table to determine actuation of the high and low side switches. The lookup table acts as a state machine and is shown in Figure 2.2.

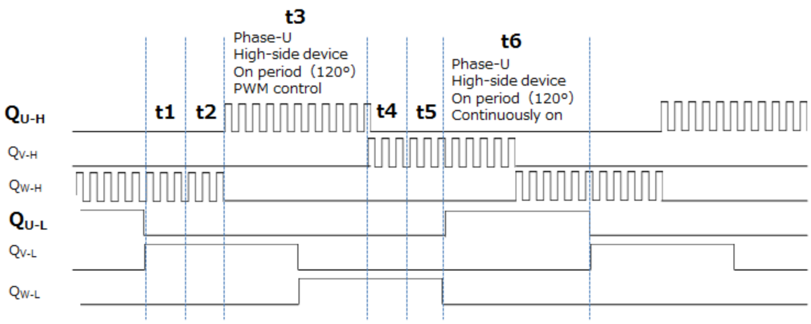
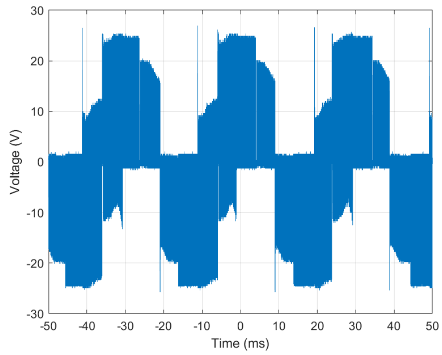


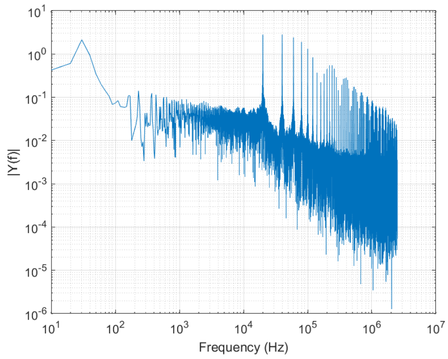
Figure 2.1. Commutation sequence for a sensored BLDC motor [2]

Input Hex Value	in3	in2	in1	in0		out5	out4	out3	out2	out1	out0	Output Hex Value
00	0	0	0	0		0	0	0	0	0	0	0x00
01	0	0	0	1		1	0	0	0	0	0	0x20
02	0	0	1	0		0	1	0	0	0	0	0x10
03	0	0	1	1		1	0	0	0	0	0	0x20
04	0	1	0	0		0	0	1	0	0	0	0x08
05	0	1	0	1		0	0	1	0	0	0	0x08
06	0	1	1	0		0	1	0	0	0	0	0x10
07	0	1	1	1		0	0	0	0	0	0	0x00
08	1	0	0	0		0	0	0	0	0	0	0x00
09	1	0	0	1		1	0	0	0	1	0	0x22
0A	1	0	1	0		0	1	0	0	0	1	0x11
0B	1	0	1	1		1	0	0	0	0	1	0x21
0C	1	1	0	0		0	0	1	1	0	0	0x0C
0D	1	1	0	1		0	0	1	0	1	0	0x0A
0E	1	1	1	0		0	1	0	1	0	0	0x14
0F	1	1	1	1		0	0	0	0	0	0	0x00

Figure 2.2. Lookup table for the commutation sequence [2]

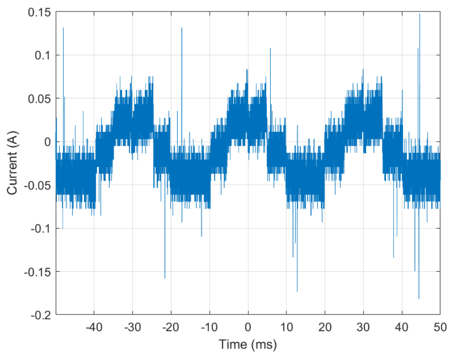


(a) BC phase to phase voltage

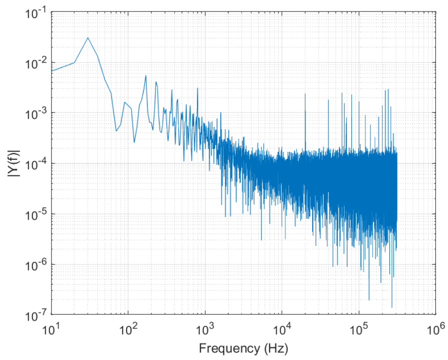


(b) FFT of the BC phase to phase voltage

Figure 2.3. Phase to phase voltage of the sensored BLDC motor



(a) Current of Phase A



(b) FFT of Current of Phase A

Figure 2.4. Current of the sensored BLDC motor

2.2 Measurements

In the fft of the phase to phase voltage, the fundamental frequency is at 30 Hz (excluding the frequency at 20kHz).

The spikes at the higher frequency, starting at 20 kHz are due to the PWM signal used to control the motor. This can be verified by looking at the code used to control the motor the clock frequency is 4800 kHz and there is a counter counting from 0 to 239 to generate one pulse. Therefore the PWM frequency is 20 kHz as shown in Equation 2.1

$$f_{\mathrm{PWM}} = \frac{4800\ \mathrm{kHz}}{240} = 20\ \mathrm{kHz} \tag{2.1}$$

The current of phase A has a fundamental frequency of 30 Hz as well. The spikes at the higher frequency are due to the PWM signal as well. For the current these are lower than for the voltage. This is due to the inductance of the motor which smoothens the current.

The RPM of the motor was also measured and was found to be 495 RPM or 8.25 Hz respectivly. As the motor has 8 poles (4 pole pairs), the electrical frequency is 4 times the mechanical frequency. Therefore the electrical frequency is 33 Hz. The discrepancy between the measured and calculated frequency is due to the fact that the resolution of the fft is not high enough to measure the exact frequency. This also explains the broad peak in the fft of the current around the fundamental frequency.

The THD was also calculated, although the calculations are not accurate as the resolution of the fft is not high enough. Zero padding was used to increase the resolution of the fft. The resolution was increased by a factor of 4 as we padded the signal with 3 times its length in zeros.

The THD_F was calculated using the formula in Equation 2.2.

$$THD_F = \frac{\sqrt{\sum_{i=2}^{5} X_i^2}}{X_1} \tag{2.2}$$

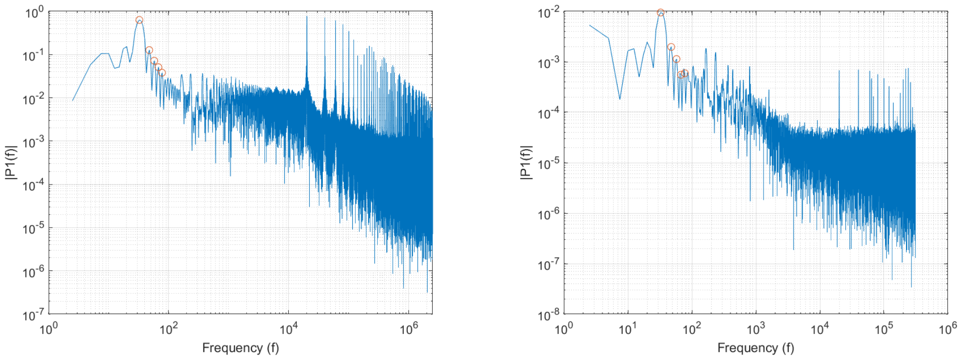
The FFT of the zero padded phase to phase voltage and the the harmonics used for the calculation of the THD are shown in Figure 2.5a.

The FFT of the zero padded current and the the harmonics used for the calculation of the THD are shown in Figure 2.5b.

The THD_F of the phase to phase voltage was found to be 25.151% and the THD of the current was found to be 25.4276%.

The meassurement were repeated for a higher speed of approximately 207 Hz electrical speed respectively 51.75 Hz mechanical speed.

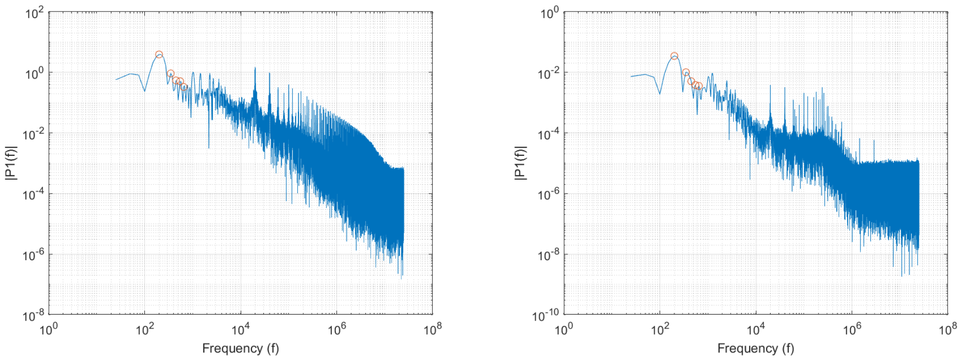
The results of the meassurement are shown in Figure 2.6. The THD_F of the phase to phase voltage was found to be 31.513% and the THD of the current was found to



(a) FFT Phase to phase with zero padding **(b)** FFT Current with zero padding

Figure 2.5. FFT of the sensored BLDC motor

be 35.727%. These THD values are higher than for the lower speed already discussed. This implies that the distortion of the current and voltage increases with the speed of the motor.



(a) FFT Phase to phase with zero padding **(b)** FFT Current with zero padding

Figure 2.6. FFT of the sensored BLDC motor

Chapter 3

Sensorless BLDC 120° Block Commutation

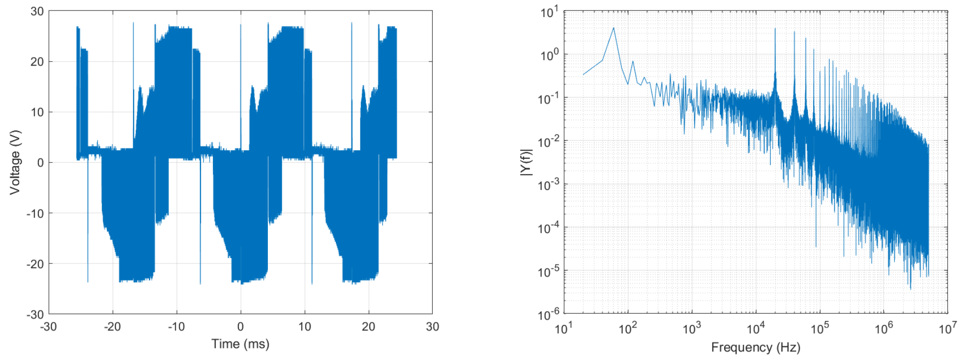
3.1 Control Strategy

Figure 3.1. Phase A voltage (green) and Hall effect sensor (yellow)

Figure 3.1 shows the BLDC's A-phase voltage in corelation to the signal acquired by the H1 hall-efect sensor. Using Figure 3.1 we can try to understand the control strategy that is used for sensorless BLDC's. Here, the falling and rising slopes of the a-phase voltage, or of all three phases for that matter, show the back-EMF induced by the BLDC's spinning rotor. Encapsulated by these slopes the actual a-phase voltage applied by the motor controller as a PWM signal can be seen.

By measuring the back-EMF of all three phases and further interpolating the data, the motor controller can then calculate the rotors position, thus making the BLDC controllable without the use of additional sensors.

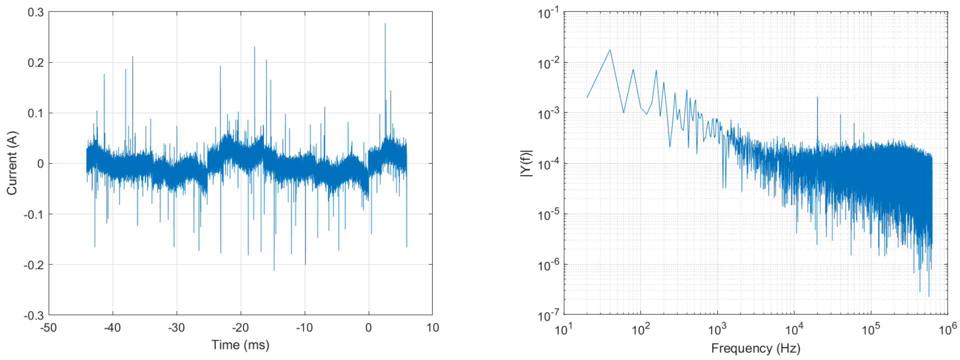
3.2 Measurements



(a) BC phase to phase voltage

(b) FFT of the BC phase to phase voltage

Figure 3.2. Phase to phase voltage of the sensorless BLDC motor



(a) Current of Phase A

(b) FFT of Current of Phase A

Figure 3.3. Current of the sensorless BLDC motor

Similar to the measurements depicted in chapter 2, a low-frequency spike, this time at around 50 Hz, and a high-frequency spike at 20 kHz can be seen. These corelate to the electrical frequency of the BLDC and the PWMs fundamental switching frequency respectively. The low-frequency peak occuring at 50 Hz can be attributed to a different speed setting

The THD_F Values for the current and voltage of the sensorless BLDC were calculated according to Equation 2.2 and were found to be 43.78% and 70.12% for the voltage and current respectively.

Chapter 4

Field Oriented Control

Field oriented control (FOC) is a control strategy for BLDC motors that allows for control of the torque and speed of the motor. The control strategy is based on the transformation of the stator current and voltage to a rotating reference frame. The transformation is done using the Park and Clarke transformation. A block diagram of the FOC control strategy is shown in Figure 4.1.

It is important to note that the FOC control strategy requires knowledge of the rotor position. This is usually done using a sensor, but here a sensorless approach is used. In this implementation the position is estimated from the currents after the clark transformation.

The main idea is use the Back EMF to estimate the angel θ of the rotor. But as the Back EMF can take high voltage values, the currents are used to estimate the rotor position.

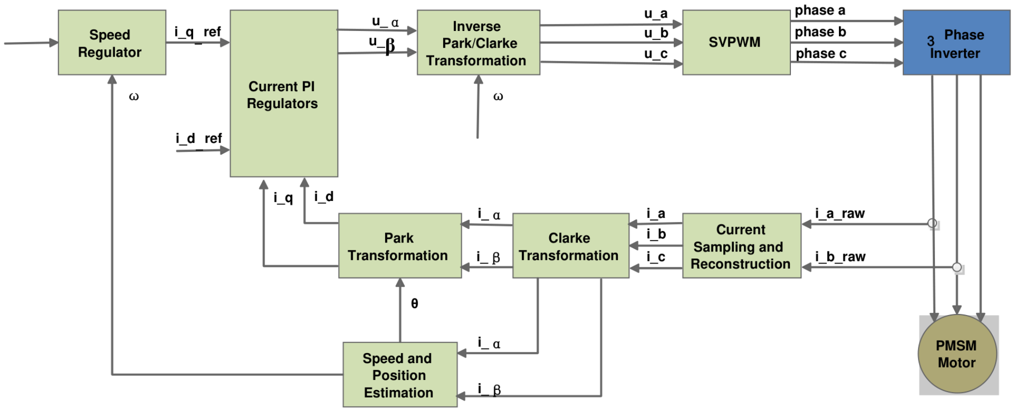
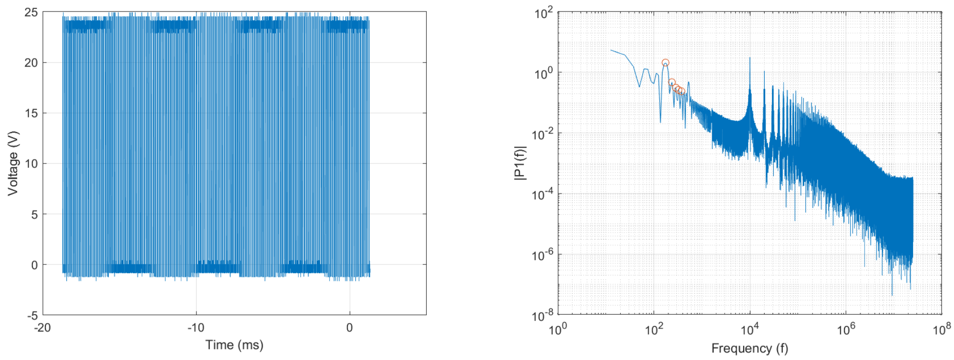


Figure 4.1. FOC Block Diagram [1]

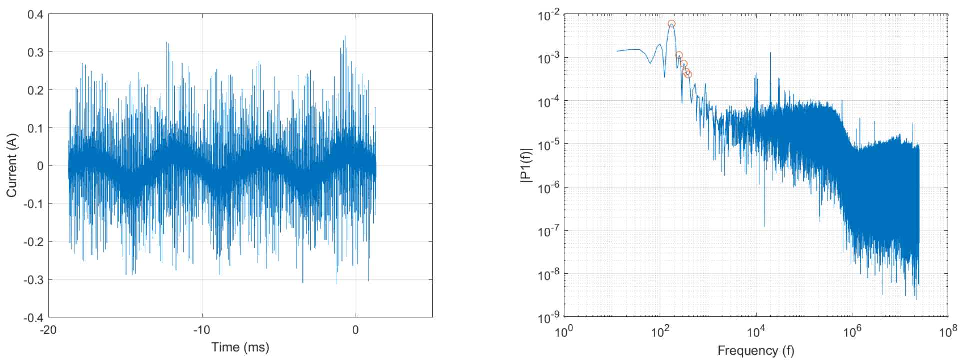
4.1 Measurements

The measurements were performed for a low speed of 8.25 Hz again. The phase voltage and current are shown in Figure 4.2a and Figure 4.3a respectively. For a better understanding, we also low pass filtered the signals and calculated the FFT of the signals. The low pass filter applied is a iir filter with a cutoff frequency of 2000 Hz.



(a) Phase voltage **(b)** FFT of the phase voltage

Figure 4.2. Phase to phase voltage of the sensored BLDC motor



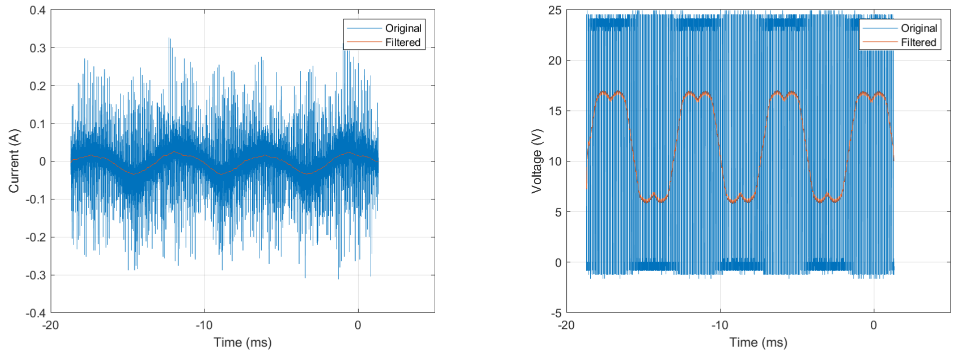
(a) Current of Phase A **(b)** FFT of Current of Phase A

Figure 4.3. Current of the sensored BLDC motor

The THD_F is calculated in the same way as described in chapter 2 The THD_F of the phase voltage was found to be 31.76% and the THD of the current was found to be 24.4%.

4.2 Limitations

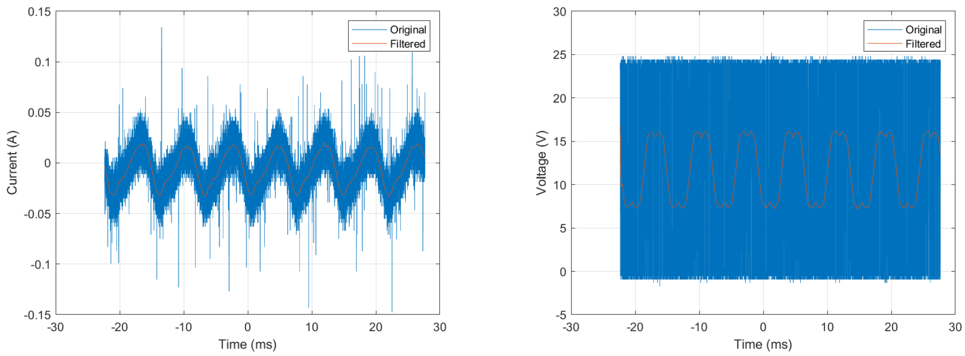
For the FOC control strategy the motor was not able to drive the motor at a higher load. This could be due to the implementation of the sensorless approach. The sensorless approach is based on the currents and the currents are not ideal for estimating the rotor position. Especialy at higher speeds the currents become more



(a) FOC Current with lowpassCurrent of Phase A **(b)** FOC Voltage with lowpass

Figure 4.4. FOC signals with low pass filter

distorted and the estimation of the rotor position becomes more difficult. This is shown in Figure 4.5.



(a) FOC Current with lowpassCurrent of Phase A **(b)** FOC Voltage with lowpass

Figure 4.5. FOC signals with low pass filter at higher speed

Bibliography

[1] Anshul Gulati. “CY8CKIT-037 PSoC® 4 Motor Control Evaluation Kit Guide”. In: 001 ().

[2] Toshiba. *application_note_en_20180803_AKX00303.Pdf*. Aug. 3, 2018. URL: https://toshiba.semicon-storage.com/info/application_note_en_20180803_AKX00303.pdf?did=61176 (visited on 05/23/2024).

List of Figures

List of Tables

Appendix A

MATLAB script

```
    clear all
clear
close all

outputDirectory = '..\img';

[timeVector1, voltageVector1] = importAgilentBin('./LabSession3Meassurements/scope_24.bin',

voltageVector1 = voltageVector1 - 0.222;

figure('Name','Phase A Current 1');
t_ms = timeVector1 * 1000;
plot(t_ms, voltageVector1);
grid on;
xticks(-50:10:50);
xlabel('Time (ms)');
ylabel('Current (A)');
saveas(gcf, fullfile(outputDirectory, 'sensored_current.png'));




% Extract the time and data from the timeseries object
t = timeVector1;
x = voltageVector1;


% Compute the sampling frequency assuming equidistant time points
Fs = 1 / mean(diff(t));

thd_db = thd(x)
percent_thd = 100*(10^(thd_db/20))
% Compute the FFT
Y = fft(x);

% Set DC component to 0 as we meassured to Uref
Y(1)=0;
```

```
% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2 = abs(Y / length(x));
P1 = P2(1:length(x)/2 + 1);
P1(2:end-1) = 2 * P1(2:end-1);

% Define the frequency domain f
f = Fs * (0:(length(x)/2)) / length(x);

% Plot single-sided amplitude spectrum
figure('Name','Single-Sided Amplitude Spectrum of A Current');
p = loglog(f, P1);
grid on;
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
saveas(gcf, fullfile(outputDirectory, 'sensored_current_fft.png'));
```

```
x = x';
% Zero-padding: extend x with zeros to make its length 4 times the original
N_original = length(x);
N_padded = 4 * N_original;
x_padded = [x zeros(1, N_padded - N_original)];

% Compute the FFT of the zero-padded signal
Y_padded = fft(x_padded);
% Y_padded(1:3) = [0,0,0]

% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2_padded = abs(Y_padded / N_padded);
P1_padded = P2_padded(1:N_padded / 2 + 1);
P1_padded(2:end-1) = 2 * P1_padded(2:end-1);

% Define the frequency domain f for the zero-padded signal
f_padded = Fs * (0:(N_padded / 2)) / N_padded;

idx = 14:4:(14+10*4);
idx = [14, 20, 24, 28, 32]
f_THD_padded = f_padded(idx);
P_THD_padded = P1_padded(idx);
% calculate THD in percent
THD_F_current = rssq(P_THD_padded(2:end))/P_THD_padded(1)*100

% (Optional) Plot the single-sided amplitude spectrum
figure;
p = loglog(f_padded, P1_padded, f_THD_padded, P_THD_padded);
grid on;
p(2).Marker='o';
p(2).LineStyle='none';
xlabel('Frequency (f)');
ylabel('|P1(f)|');

saveas(gcf, fullfile(outputDirectory, 'sensored_current_fft_padded.png'));
```

```
[timeVector1, voltageVector1] = importAgilentBin('./LabSession3Meassurements/scope_25.bin',
[timeVector2, voltageVector2] = importAgilentBin('./LabSession3Meassurements/scope_25.bin',

bcvoltage = voltageVector1-voltageVector2;
figure('Name','Phase B-C');
t_ms = timeVector1 * 1000;
plot(t_ms, bcvoltage);
grid on;
xticks(-50:10:50);
xlabel('Time (ms)');
ylabel('Voltage (V)');
saveas(gcf, fullfile(outputDirectory, 'sensored_voltage.png'));
% Extract the time and data from the timeseries object
t = timeVector1;
x = bcvoltage ;

% Compute the sampling frequency assuming equidistant time points
Fs = 1 / mean(diff(t));

% Compute the FFT
Y = fft(x);

% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2 = abs(Y / length(x));
P1 = P2(1:length(x)/2 + 1);
P1(2:end-1) = 2 * P1(2:end-1);

% Define the frequency domain f
f = Fs * (0:(length(x)/2)) / length(x);
```

```
x = x';
% Zero-padding: extend x with zeros to make its length 4 times the original
N_original = length(x);
N_padded = 4 * N_original;
x_padded = [x zeros(1, N_padded - N_original)];

% Compute the FFT of the zero-padded signal
Y_padded = fft(x_padded);

% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2_padded = abs(Y_padded / N_padded);
```

```
P1_padded = P2_padded(1:N_padded / 2 + 1);
P1_padded(2:end-1) = 2 * P1_padded(2:end-1);

% Define the frequency domain f for the zero-padded signal
f_padded = Fs * (0:(N_padded / 2)) / N_padded;

idx = 14:4:(14+10*4);
idx = [14, 20, 24, 28, 32]
f_THD_padded = f_padded(idx);
P_THD_padded = P1_padded(idx);
% calculate THD in percent
THD_F_voltage = rssq(P_THD_padded(2:end))/P_THD_padded(1)*100

% (Optional) Plot the single-sided amplitude spectrum
figure;
p = loglog(f_padded, P1_padded, f_THD_padded, P_THD_padded);
grid on;
p(2).Marker='o';
p(2).LineStyle='none';
xlabel('Frequency (f)');
ylabel('|P1(f)|');

saveas(gcf, fullfile(outputDirectory, 'sensored_voltageBC_fft_padded.png'));


% Plot single-sided amplitude spectrum
figure;
title('Single-Sided Amplitude Spectrum of BC Phase');
p = loglog(f, P1);
grid on;
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');

saveas(gcf, fullfile(outputDirectory, 'sensored_voltageBC_fft.png'));
```

```
[timeVector1, voltageVector1] = importAgilentBin('./LabSession3Meassurements/scope_29.bin',

voltageVector1 = voltageVector1 - 0.372;

figure('Name','Phase A Current 1');
t_ms = timeVector1 * 1000;
plot(t_ms, voltageVector1);
grid on;
xticks(-50:10:50);
xlabel('Time (ms)');
ylabel('Current (A)');
saveas(gcf, fullfile(outputDirectory, 'sensored_speed_current.png'));
```

```
% Extract the time and data from the timeseries object
t = timeVector1;
x = voltageVector1;


% Compute the sampling frequency assuming equidistant time points
Fs = 1 / mean(diff(t));

thd_db = thd(x)
percent_thd = 100*(10^(thd_db/20))
% Compute the FFT
Y = fft(x);

% Set DC component to 0 as we meassured to Uref
Y(1)=0;

% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2 = abs(Y / length(x));
P1 = P2(1:length(x)/2 + 1);
P1(2:end-1) = 2 * P1(2:end-1);

% Define the frequency domain f
f = Fs * (0:(length(x)/2)) / length(x);

% Plot single-sided amplitude spectrum
figure('Name','Single-Sided Amplitude Spectrum of A Current');
p = loglog(f, P1);
grid on;
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
saveas(gcf, fullfile(outputDirectory, 'sensored_speed_current_fft.png'));
```

```
x = x';
% Zero-padding: extend x with zeros to make its length 4 times the original
N_original = length(x);
N_padded = 4 * N_original;
x_padded = [x zeros(1, N_padded - N_original)];

% Compute the FFT of the zero-padded signal
Y_padded = fft(x_padded);
% Y_padded(1:3) = [0,0,0]

% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2_padded = abs(Y_padded / N_padded);
P1_padded = P2_padded(1:N_padded / 2 + 1);
P1_padded(2:end-1) = 2 * P1_padded(2:end-1);

% Define the frequency domain f for the zero-padded signal
f_padded = Fs * (0:(N_padded / 2)) / N_padded;

idx = 14:4:(14+10*4);
idx = [9, 15, 19, 23, 27]
f_THD_padded = f_padded(idx);
P_THD_padded = P1_padded(idx);
% calculate THD in percent
THD_F_current = rssq(P_THD_padded(2:end))/P_THD_padded(1)*100

% (Optional) Plot the single-sided amplitude spectrum
figure;
p = loglog(f_padded, P1_padded, f_THD_padded, P_THD_padded);
grid on;
p(2).Marker='o';
p(2).LineStyle='none';
xlabel('Frequency (f)');
ylabel('|P1(f)|');

saveas(gcf, fullfile(outputDirectory, 'sensored_speed_current_fft_padded.png'));
```

```
[timeVector1, voltageVector1] = importAgilentBin('./LabSession3Meassurements/scope_31.bin',
[timeVector2, voltageVector2] = importAgilentBin('./LabSession3Meassurements/scope_31.bin',

bcvoltage = voltageVector1-voltageVector2;
figure('Name','Phase B-C');
t_ms = timeVector1 * 1000;
```

```
plot(t_ms, bcvoltage);
grid on;
xticks(-50:10:50);
xlabel('Time (ms)');
ylabel('Voltage (V)');
saveas(gcf, fullfile(outputDirectory, 'sensored_speed_voltage.png'));
% Extract the time and data from the timeseries object
t = timeVector1;
x = bcvoltage ;

% Compute the sampling frequency assuming equidistant time points
Fs = 1 / mean(diff(t));

% Compute the FFT
Y = fft(x);

% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2 = abs(Y / length(x));
P1 = P2(1:length(x)/2 + 1);
P1(2:end-1) = 2 * P1(2:end-1);

% Define the frequency domain f
f = Fs * (0:(length(x)/2)) / length(x);




x = x';
% Zero-padding: extend x with zeros to make its length 4 times the original
N_original = length(x);
N_padded = 4 * N_original;
x_padded = [x zeros(1, N_padded - N_original)];

% Compute the FFT of the zero-padded signal
Y_padded = fft(x_padded);

% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2_padded = abs(Y_padded / N_padded);
P1_padded = P2_padded(1:N_padded / 2 + 1);
P1_padded(2:end-1) = 2 * P1_padded(2:end-1);

% Define the frequency domain f for the zero-padded signal
f_padded = Fs * (0:(N_padded / 2)) / N_padded;

idx = 14:4:(14+10*4);
idx = [9, 15, 19, 23, 27]
f_THD_padded = f_padded(idx);
P_THD_padded = P1_padded(idx);
% calculate THD in percent
THD_F_voltage = rssq(P_THD_padded(2:end))/P_THD_padded(1)*100

% (Optional) Plot the single-sided amplitude spectrum
figure;
p = loglog(f_padded, P1_padded, f_THD_padded, P_THD_padded);
grid on;
p(2).Marker='o';
```

```
p(2).LineStyle='none';
xlabel('Frequency (f)');
ylabel('|P1(f)|');

saveas(gcf, fullfile(outputDirectory, 'sensored_speed_voltageBC_fft_padded.png'));


% Plot single-sided amplitude spectrum
figure;
title('Single-Sided Amplitude Spectrum of BC Phase');
p = loglog(f, P1);
grid on;
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');

saveas(gcf, fullfile(outputDirectory, 'sensored_speed_voltageBC_fft.png'));
```