



CHALMERS
UNIVERSITY OF TECHNOLOGY



Playing music with an permanent-magnet synchronous motor

Master's thesis in Electrical power engineering

Johan Olsson

DEPARTMENT OF Electrical Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

MASTER'S THESIS 2022

Playing music with an permanent-magnet synchronous motor

JOHAN OLSSON



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Electrical Power Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022

Playing music with an permanent-magnetsynchronous motor

Johan Olsson

© JOHAN OLSSON, 2022.

Supervisor: Joachim Härsjö, Volvo Cars

Examiner: Stefan Lundberg, Electrical Engineering

Master's Thesis 2022

Department of Electrical Engineering

Division of Electrical Power Engineering

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Typeset in L^AT_EX

Printed by Chalmers Reproservice

Gothenburg, Sweden 2022

Johan Olsson
Department of Electrical engineering
Chalmers University of Technology

Abstract

Electric vehicles become more common and since they have no internal combustion engine, they are much quieter, which causes more accidents involving pedestrians and cyclists. Demands are made by authorities that electric vehicles should make some sound when traveling below some speed limit. In some electric vehicles the permanent magnet synchronous machine (PMSM) is used as traction motor. Due to a physical phenomenon called magnetostriction the PMSM can be temporarily deformed, and that causes the PMSM to make a sound.

The main goal of the thesis was to play music with the PMSM without impacting the average torque production. This was done by testing two different methods that was believed to create sound. The first method superimposed a voltage on top on the voltage that are supplied to the machine. The second method puts all transistors in a non-conductive state in the inverter to create a voltage dip and thereby a resulting current dip. It was found that it is possible to play music on an PMSM with both methods without impacting the average torque, within the boundaries of measurement error, and that the frequency response of the PMSM depend on which method that is used.

It was not possible to only generate the requested frequency with the evaluated methods. The sound produced by the motor always contained multiple frequencies. For the stop-switching method the requested frequency was always present in the sound produced by the motor, but it was not always the frequency with the highest intensity. For the superimpose method the requested frequency was present in most cases but not in all and it was not always the frequency with the highest intensity. But when measuring the acoustic transfer function of the lab-rig PMSM, i.e. when increasing the requested frequency in steps, the frequency response from the motor creates an audible difference between the requested frequencies when listening and a visible distance of change in frequency spectrum when recorded and analyzed. This makes it possible for the listener to identify the song played, even if the motor does not play exactly the right frequency.

Keywords: PMSM, Music, Magnetostriction,

Acknowledgements

I would like to express my sincere gratitude to my supervisor at Volvo Cars Joachim Härsjö for his patience, encouragement and continuous support of my thesis.

I would also like to thank Rasmus Hofwimmer and Mikael Lindholm for their technical guidance, which was necessary to be able to carry out my thesis.

Finally, I would like to thank examiner and supervisor at Chalmers University of Technology, Stefan Lundberg for the feedback throughout the thesis.

Johan Olsson, Gothenburg, June 2021

Contents

List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Background	1
1.2 Aim	2
1.3 Scope	2
2 Sound generation in an electric power-train	3
2.1 Sound	3
2.2 Analogue signals	5
2.3 Electromagnetism	7
2.3.1 Ampère’s law	7
2.3.2 Faradays’s law of induction	8
2.3.3 Magnetic fields and forces	9
2.3.4 Magnetostriction	10
2.4 Electric Powertrain	12
2.4.1 Three-phase inverter	12
2.4.2 Electric machines	15
2.4.3 PMSM circuit model	17
2.4.4 Dynamic modeling of the PMSM	18
2.5 Electric machine controller	22
2.6 Implementation of the different methods	23
3 How to generate sound with an electric powertrain	25
3.1 Test rigs	25
3.2 Superimpose	26
3.2.1 Tone creation for superimpose method	28
3.2.2 Implementation of superimpose solution	30
3.3 Stop-switching method	32
3.3.1 Implementation of the stop-switching solution	33
3.4 Acoustic transfer function of the lab rig PMSM	35
4 Measured sound from the test rigs	37
4.1 Measurements using the superimpose method	37
4.1.1 Results from lab rig	37

4.1.2	Results from production rig	41
4.2	Measurements using the stop-switching method	43
4.3	Acoustic transfer function of the lab-rig PMSM	48
5	Conclusion	57
5.1	Future work	58
	Bibliography	59
A	Appendix 1	I
A.1	Code	I

List of Figures

2.1	Three signals with the same fundamental period and different degree of superimposed frequencies	5
2.2	The effect of aliasing	6
2.3	Current through conductor with resulting magnetic field ("electromagnetism" by Stannered is licensed under CC BY-SA 3.0)	7
2.4	Flux linkage between two coils	8
2.5	Permeability of a ferromagnetic material and vacuum subject of the same magnetic field	9
2.6	The effect of magnetostriction	11
2.7	Cross section of an PMSM with flux lines Adopted from [14]	12
2.8	Deforming of the stator when there is a sinusoidal current running through the stator windings.	12
2.9	Three phase inverter without gate drivers	13
2.10	Switching pattern for first sector	14
2.11	The rotor of an electric machines follows the magnitude of the current	16
2.12	RL-circuit	17
2.13	PMSM-circuit for one phase when the rotor is rotating	17
2.14	Alpha-beta to DQ system	19
2.15	PMSM three-phase equations	19
2.16	circuit representation of u_{sd} and u_{sq}	21
2.17	General model over the controller used	22
2.18	The superimposed signal are applied to the D-voltage inside the controller	23
2.19	The stop-switching method is implemented in the PWM-modulator block to set gate drivers in passive zero	24
3.1	The small test rig	26
3.2	Output from different counters	28
3.3	Train by Karl Lindblom	29
3.4	Difference between dynamic and constant sampling	30
3.5	Implementation of the superimpose method as a block diagram	31
3.6	Implementation of the stop-switching method as a block diagram	34
3.7	SQuadriga Recording equipment inside rig	36
4.1	Rotor speed and current measurements when a rotational case was performed and no music was added to the system.	38

4.2	Electric rotor speed with superimposed song when 3A were requested in D and Q currents	39
4.3	PWM with song	40
4.4	PWM-pattern from inverter to the machine when song is superimposed	40
4.5	Mechanical rotor speed from the machine from the non-rotational case when song is superimposed on U_D	41
4.6	Mechanical rotor speed from the machine when the rotational case was performed and a song was superimposed on U_D	42
4.7	Dynamic gain function	43
4.8	Comparison for the rotating case when stop-switch method was performed.	45
4.9	PWM-pattern from one phase leg with song playing	47
4.10	The acoustic transfer function of the PMSM when using stop-switching method and the PMSM is stationary	48
4.11	Frequency spectrum of the recorded sound form the PMSM when 500 Hz and 700 Hz were requested.	49
4.12	Result from the SQadriga measurements analyzed with the PAK software for the stop-switching method playing single frequencies and the PMSM is non-rotating. The requested frequencies are 100, 200, 303, 400, 500, 625, 714 and 800 Hz for 10 s each (0-10 s 100 Hz,..... 70-80 s 800 Hz)	50
4.13	Result from the SQadriga measurements analyzed with the PAK software for the stop-switching method playing single frequencies and the PMSM is non-rotating. The requested frequencies are 909, 1000, 1250, 1428, 2000, 2500, 3333 5000 Hz for 10 s each (80-90 s 909 Hz,..... 160-170 s 5000 Hz).	51
4.14	Result from the SQadriga measurements analyzed with the PAK software for the stop-switching method playing single frequencies and the PMSM is non-rotating. The requested frequencies are 100, 200, 303, 400, 500, 625, 714 and 800 Hz for 10 s each (0-10 s 100 Hz,..... 70-80 s 800 Hz)	52
4.15	Result from the SQadriga measurements analyzed with the PAK software for the stop-switching method playing single frequencies and the PMSM is non-rotating. The requested frequencies are 909, 1000, 1250, 1428, 2000, 2500, 3333 5000 Hz for 10 s each (80-90 s 909 Hz,..... 160-170 s 5000 Hz).	53
4.16	Result from the SQadriga measurements analyzed with the PAK software for the superimpose method playing single frequencies and the PMSM is non-rotating. The requested frequencies are shown in Table 3.4 and they are played for 5 s each (0-5 s 100 Hz, 5-10 s 200 Hz,....., 70-75 s 3.3 kHz) and in addition 5 kHz is played between 75 s and 80 s.	55

List of Tables

3.1	Frequency and time vectors as they are stored in the look-up table . .	31
3.2	Resolution becomes higher among lower frequencies due to the inverse relationship between time and frequency	32
3.3	Time periods for chosen frequencies and corresponding desired gain value	33
3.4	Counter value and what frequency that results in	35

1

Introduction

1.1 Background

Global warming is one of humanity's greatest challenges and to deal with this problem world leaders have signed the Paris agreement. The Paris agreement's goal is to limit global warming to a maximum of 2 degrees Celsius, but preferable lower. Every country must do its best to fulfill this task since we're all in it together. Different steps are taken by each country to meet the requirements and one step that each country eventually must take is the transition to vehicles that run on fossil free energy, of which one type is the electric vehicles [1].

With new technological solutions come new problems since vehicles without combustion engines, such as electric vehicles, make less noise when traveling. This is a preferred trait of electric vehicles since it lowers the overall noise level in a city but it comes with an unexpected drawback. The lower noise levels are causing trouble when pedestrians in cities are crossing roads, especially pedestrians with partial hearing or sight [2]. A study from the US department of transportation National Highway Traffic Safety Administration showed in 2011 that the likelihood of getting hit by an electric or hybrid vehicle was 37% higher for pedestrians and 57% higher for cyclists compared to a vehicle with a combustion engine [3]. Because of this many countries demand that electric vehicles must make an artificial sound when traveling below a certain speed limit to create awareness of the vehicles. Today this is done by placing speakers outside the car to play some kind of sound to make it easier for pedestrians to notice that the car approaches [4].

Another possible solution is to use the existing hardware that comes with an electric vehicle, the electric motor. It is possible to use the electric motor as a speaker, due to a phenomenon called magnetostriction [5]. By taking advantage of this phenomenon it is possible, in a controlled way to create sounds with the electric motor. Playing an artificial speed sound or music when driving at low speeds, creating awareness of an approaching vehicle. Thus reducing the risk of those pedestrians and cyclists being hit by a car.

1.2 Aim

The thesis aims to develop at least one method to play music with a permanent magnet synchronous machine by manipulating the machine currents by controlling the inverter output while at the same time not affecting the average torque to the drive system.

1.3 Scope

The controller is implemented and simulated in MATLAB/ SIMULINK. Two different methods are evaluated; The first method, where the frequencies of the sound is superimposed on the output signal from the controller, and the second where solid-state switching method is used. The different methods are tested on a smaller type of drive train provided by Volvo Cars before it's implemented on the electric machine currently used in Volvos' electric cars.

2

Sound generation in an electric power-train

To play music with a Permanent Magnet Synchronous Motor(PMSM) two methods are tested. Both methods takes advantage of a physical phenomena called magnetostriction. The first method is to superimpose an additional voltage with same frequencies as in the music on top of the voltage required for normal operation of the machine, hence the name superimpose method.

The other method is to set all gate drivers in the inverter in a non-conductive state for a short period of time, which forces all IGBTs to stop conducting current to the machine during this time. That creates a bump in the current causing harmonics to arise, by doing this in a controlled way, the harmonics that arise are controlled in a way so that the machine generate the desired acoustic sound.

To play music with a PMSM first some general theory regarding of what sound is, the physical phenomena called magnetostriction which is be used to create sound with the PMSM, some information about the parts that are responsible for the sound creation, and how the creation of sound can be integrated into a modern PMSM controller will be presented.

2.1 Sound

Sound is most commonly experienced as waves traveling through air and it results in the human perception of hearing. When sound waves travel through any medium the elements are disturbed from their original position. The propagation of sound waves changes the pressure and density in the medium along the direction of the wave. Thus, if a source would create a sinusoidal vibration, the pressure and density would also vibrate sinusoidally. The sound is created by a source that expands in at least one direction, increasing the pressure and density in that direction, then the source stops, while the pulse continues, the source then contracts which creates an empty space, followed by a stop and then expansion again and by that creating another pulse [6]. The distance between the two compression peaks is the wavelength λ , it takes some time for the wavelength to complete this and the period time is called T . With the period time the frequency of the wave can be calculated by:

$$f = \frac{1}{T} \tag{2.1}$$

When music is produced the frequency that the artist wants to be played is what is commonly referred to as a note. Multiple notes are written down and distanced from each other and music is composed. Music, therefore becomes an art form in which pauses and frequencies are organized in time. The note tells the musician what frequency and how long that frequency should be played. Each note therefore have a corresponding frequency. Hence, when a note is played on an instrument something within or on the instrument is vibrating with the corresponding frequency. The physical structure of the instrument gives the note its unique touch, which by extension gives each instrument its own transfer function. A transfer function describes the relation between a specific input to a specific output for a given system.

To reproduce the music that is created speaker elements are usually used and speaker elements also have transfer functions. In this thesis the speaker element is the PMSM. In order to find out the PMSMs acoustic transfer function a predetermined set of frequencies should be played on the PMSM and the resulting sound should be recorded with some recording equipment. To measure sound comes with challenges and both noise from the measurement tool and surrounding noise can affect the measurement to a different degree, which can be problematic.

2.2 Analogue signals

A signal is a physical quantity that can be described with a mathematical function or a table. For a periodic signal, the following condition applies: $y(t + nT) = y(t)$ where n is a integer and for any t . T is the signal period and the relation is described by (2.1). The different signals in Figure 2.1 all have the same fundamental period, T . However, there are more frequencies embedded in $y_2(t)$ and $y_3(t)$, those additional frequencies are called harmonics. If these signals would be played on a speaker element a human could distinguish between these signals because they would sound differently since some of them consist of more than one frequency. To find out which frequencies that are embedded in a analogue signal, a Fourier transform can be used. A Fourier transform move from time domain into the frequency domain and hence, a musical chord would be decomposed into its fundamental frequencies, amplitude and phase.

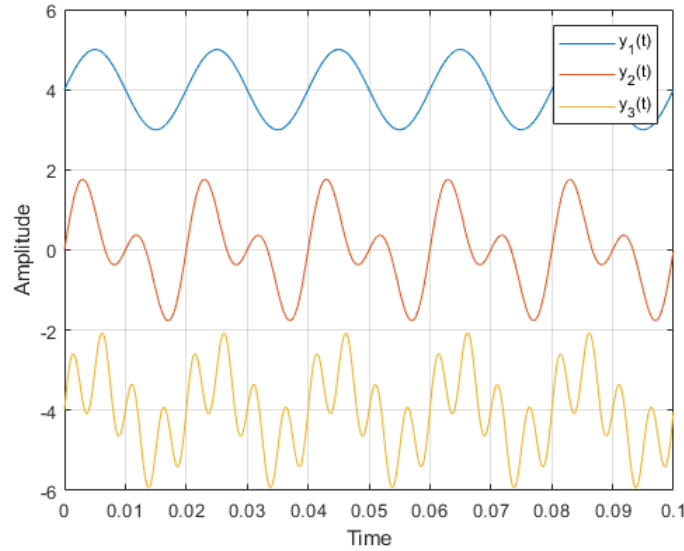


Figure 2.1: Three signals with the same fundamental period and different degree of superimposed frequencies

As can be noted, the signals in Figure 2.1 are continuous signals. However, since most of the Fourier transforms is done on computer measured signals and when a signal is measured by a computer it takes samples of the signal. The sampled signal results in a discrete function with a value for each sample. A normal Fourier transform can not be done on a discrete signal, instead a discrete Fourier transform (DFT) is performed. Solving Fourier transforms are computationally heavy and therefore an algorithm called FFT (fast Fourier transform) have been developed, the FFT algorithm solves DFTs [19]. The sampling frequency, F_s , is how many samples that are taken in one second. According to the Nyquist–Shannon sampling theorem, a signal needs to be sampled by at least twice the frequency of the signal itself in order to be recreated [20]. Otherwise, aliasing can occur. Aliasing is when the sampling frequency is too low to sample a signal correctly, a signal of higher frequency will then look like some other signal. In Figure 2.2 one can see a signal of

50 Hz is sampled with 60 Hz and the sampled signal now results in a 10 Hz signal instead of 50 Hz.

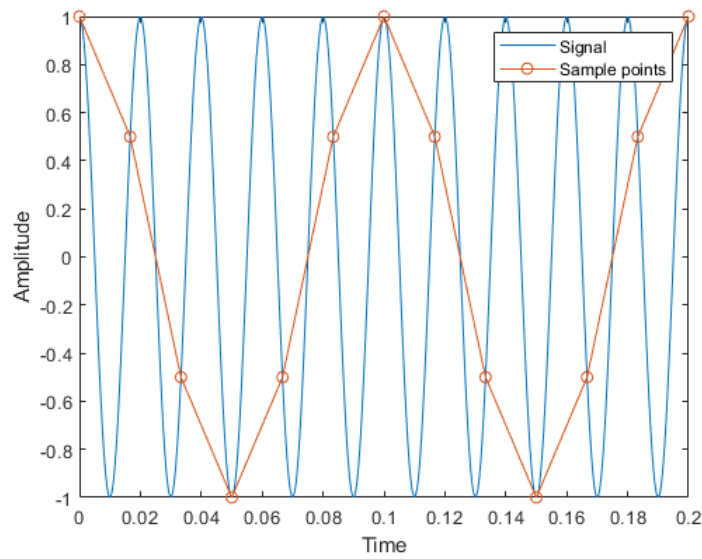


Figure 2.2: The effect of aliasing

As mentioned, measurement noise is also an issue when sampling a quantity. If a noisy quantity undergoes an FFT the result won't be the measured signal but also the result of the noise, which can be the dominating part of the measurement and thus be significantly higher than the quantity of interest.

2.3 Electromagnetism

Electromagnetism is one of physics's four fundamental forces and explains the forces between electrical charges [6]. Magnetic and electric forces are found in magnetic and electric fields. The relation between magnetism and electricity is that magnetic forces are produced when an electric charge is moving and vice versa, a changing magnetic field will produce an electric field. These phenomenons were discovered by Michael Faraday and James Maxwell respectively.

2.3.1 Ampère's law

Ampère's law states the formation of a resulting magnetic field when a continuous current is running through a conductor of any shape [6]. The law can be stated as:

$$\oint \vec{B} \cdot d\vec{s} = \mu_0 I \quad (2.2)$$

where $\oint \vec{B}$ is the line integral of the magnetic field, $d\vec{s}$ is any arbitrary closed path, I is the total current that is passing through any open surface bounded by the closed path. This means that any current that is running through a conductor creates a magnetic field that will be circulating around the moving electric charge in the direction in accordance to the right-hand rule. The field circulates the conductor on a plane perpendicular to the conductor as can be seen in Figure 2.3

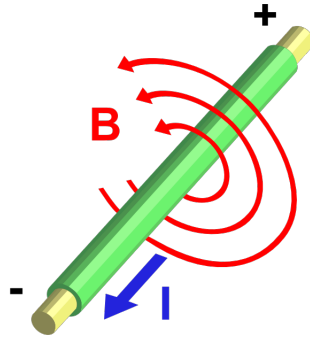


Figure 2.3: Current through conductor with resulting magnetic field
("electromagnetism" by Stannered is licensed under CC BY-SA 3.0)

2.3.2 Faradays's law of induction

Faradays's law describes how electric current can be induced in a closed loop by a changing magnetic field. That current is called induced current and it is created by the induced EMF that comes from the changing magnetic field. The loop will now behave like as if a emf source were connected to it, one can say that there is an emf produced in the loop. In general, the emf is proportional to the time derivative of the magnetic flux through a loop, in mathematical terms it can be expressed as:

$$\varepsilon = -\frac{d\Phi_B}{dt} \quad (2.3)$$

$$\Phi_B = \int \vec{B} \cdot d\vec{A} \quad (2.4)$$

Where Φ_B is magnetic flux and B is the magnetic flux density that is passing through the cross-section area of the enclosed loop. If a loop has N turns, the area remains the same and the flux is the same in each turn.

$$\Psi = N\Phi_B \quad (2.5)$$

The term Ψ is referred to as flux linkage. Flux linkage describes how many turns the flux links with, a common scenario can be seen in Figure 2.4, were not all magnetic flux that are created in one loop passes through the second loop. The amount of the flux from the first one, that also passes through the second one is called flux linkage. The negative sign in (2.3) comes from Lenz's law, which says that the induced current in a loop will oppose the change of the magnetic field [6].

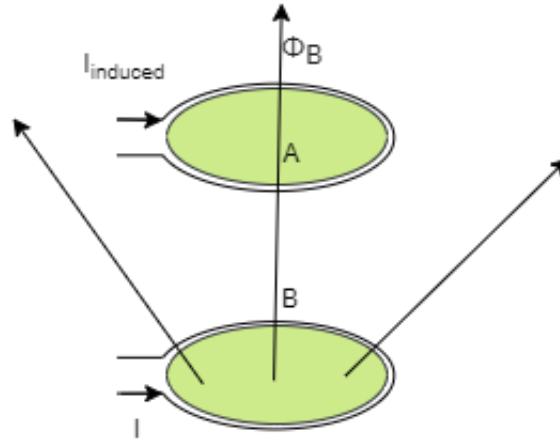


Figure 2.4: Flux linkage between two coils

2.3.3 Magnetic fields and forces

Moving electrons produces a magnetic field. Since all atoms have circulating electrons around their nucleus every atom acts as a little magnet, called magnetic dipoles. However, there is a criterion for atoms to create magnetic force, if an element has an even amount of electrons circulating in opposing directions the net magnetic field from the atom is zero and therefore are not producing any net magnetic field. On the other hand, if an element has an odd number of electrons in the outer shell such as iron, nickel, and cobalt, then the net magnetic field is no longer zero, these materials can easily be magnetized. There are two main ways to magnetize materials: one is by using permanent magnets and the other one is using an electromagnet [11].

Different materials respond differently to exposure of magnetic fields, how responsive a material is to an applied magnetic field depends on the permeability of the material, a material with higher permeability requires less magnetic field strength to achieve a set level of magnetization. A simplified example can be seen in Figure 2.5 where μ_f is the permeability of a ferromagnetic material and μ_0 is the permeability of vacuum. The H-axis is field strength and the B-axis is level of magnetization. When placing a piece of ferromagnetic material, which in general has high permeability, close to a magnetic source the dipoles of the material will align themselves with the magnetic field. This will happen quite rapidly even if the source generates a rather weak magnetic field as also can be seen in Figure 2.5. An increase in the magnetic field strength increases the alignment of the dipoles until all dipoles are aligned and then the ferromagnetic material enters what is called magnetic saturation. When a ferromagnetic material is fully saturated it does not matter how much the magnetic field strength increases, the ferromagnetic material won't be more magnetized since there are no more dipoles to align [11].

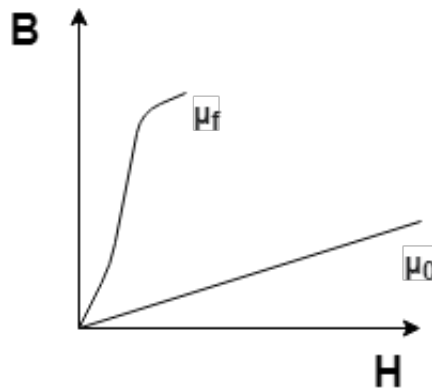
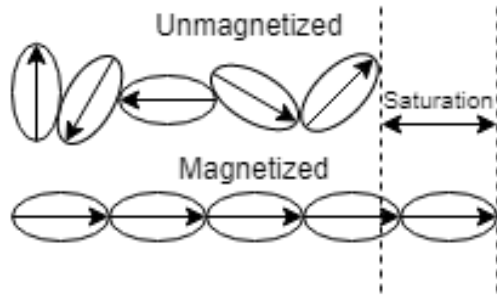


Figure 2.5: Permeability of a ferromagnetic material and vacuum subject of the same magnetic field

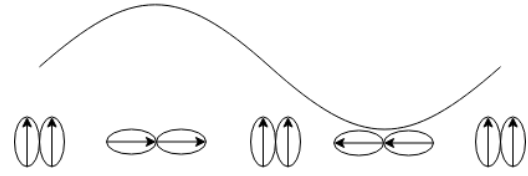
In motor applications there are two ways of magnetizing the machine, either by permanent magnets or by electromagnets. The advantage of using permanent magnets is that there is no need for electrical power to operate them, they need no maintenance and they have high efficiency. However, it's hard to control their level of magnetization, which is possible in electromagnets. Also, electromagnets can usually withstand much higher temperatures without losing magnetic strength compared to permanent magnets, but they need electrical power to operate [11].

2.3.4 Magnetostriction

When a magnetic field is applied to a material that can be magnetized the magnetic dipoles, which is in random order when the material is unmagnetized, starts to align themselves into the same direction as the magnetic field lines. This makes the material slightly longer in the direction of magnetization, as can be seen in Figure 2.6 (a). The material grows in physical length, when doing so it moves the surrounding air and thereby generates sound as described in Section 2.1. If a conductor carries current a resulting magnetic field will arise according to (2.2) and if that current is alternating the resulting magnetic field will also be alternating. The alternating magnetic field would realign the magnetic dipoles in a continuous manner in pace with the frequency of the current. As can be seen in Figure 2.6 (b) the sound generated will be twice the electrical frequency since the material from the start of the sinusoidal signal will have its original shape, then it will grow in length with the magnetic field strength, reach a peak, compress to the original size and then grow again during the negative part of the signal [12].



(a) Magnetised material grows in length



(b) Oscillates twice the frequency

Figure 2.6: The effect of magnetostriction

PMSMs are always subject of magnetic fields since they have magnets embedded or attached at the rotor and it is the magnetic fields that is creating the torque. The cross section of a PMSM as can be seen in Figure 2.7 (a). The blue yellow and green squares in the stator are windings from each phase and the blue and red larger squares are permanent magnets embedded in the rotor. In 2.7 (b) the flux lines from the magnets are seen. When the windings in the stator starts to carry current the resulting magnetic field will cause magnetostriction. However, the usual sinusoidal waves are there to create torque and not sound, therefore a new signal have to be added to create the sound. There are many ways of doing that but in this thesis, as mentioned in the introduction, two methods are tried to achieve the effect of playing music on the PMSM. The two methods alters with the voltage that is supplied to the machine and by that cause more disturbance and thereby some sound by deforming the stator slightly. An exaggerated result of the effect can be seen in Figure 2.8, where the stator is round at the initial state and is deformed in different directions depending on the direction of the magnetic field.

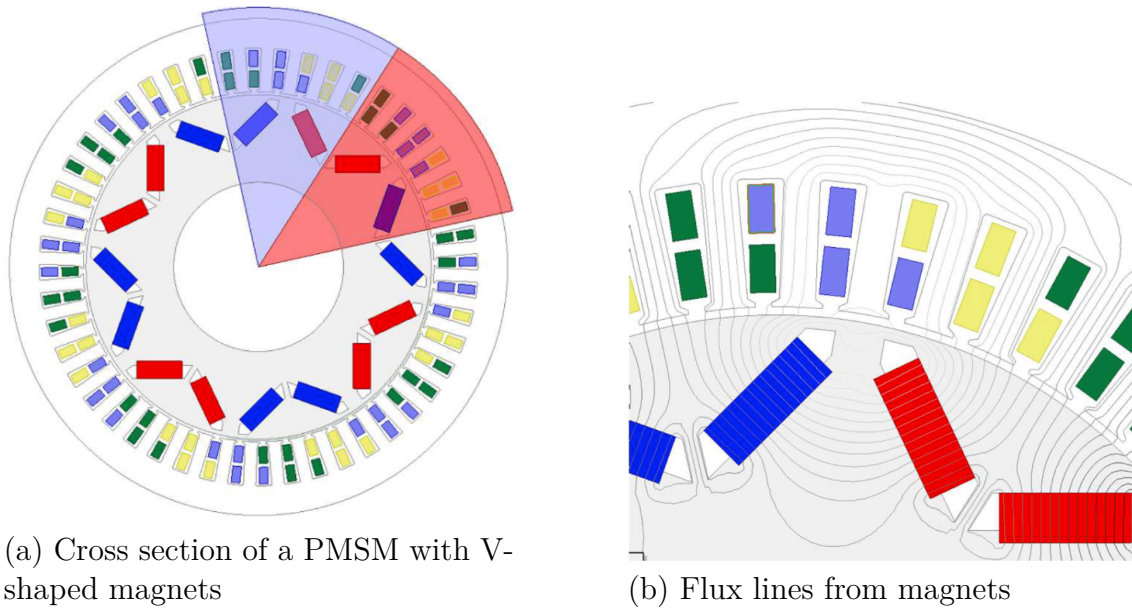


Figure 2.7: Cross section of an PMSM with flux lines
Adopted from [14]

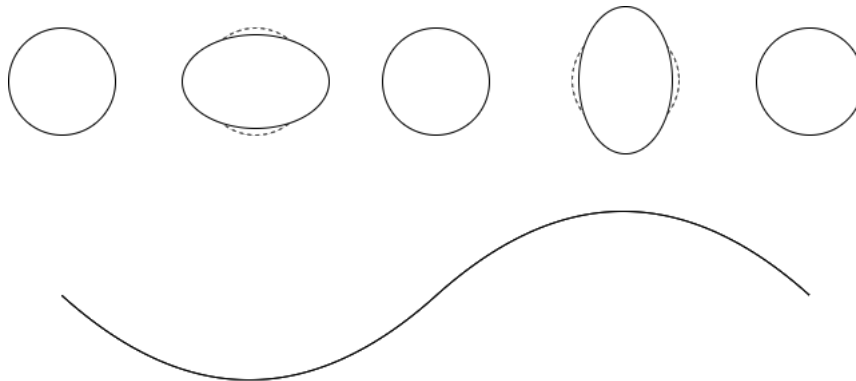


Figure 2.8: Deforming of the stator when there is a sinusoidal current running through the stator windings.

2.4 Electric Powertrain

The powertrain of an electric vehicle (EV) is all the components that are converting the power from the battery into torque. An EV's powertrain is made up of a few larger components: A battery pack, an inverter, an electric machine, and a mechanical transmission. The parts that are relevant to the thesis are presented in this section.

2.4.1 Three-phase inverter

The three-phase inverter is the device that converts DC to AC. The power electronic components in an inverter are typically; a capacitor, diodes, six transistors and gate drivers. Figure 2.7 shows the three-phase inverter, in this figure the gate drivers

are omitted. There are many kinds of transistors but in automotive applications, Insulated-Gate Bipolar Transistors (IGBTs) are common because of their ability to withstand high voltage while still being able to work at a relative high switching frequency [9]. The purpose of the capacitor is that it maintains a high voltage when there is high current demand from the electric machine. The diodes are there to provide a path for inductive current. The transistors are the switches and are controlled by the gate drivers, the gate driver in turn gets low power signals from a CPU and amplifies them to a high power signal capable of turning on and off IGBTs.

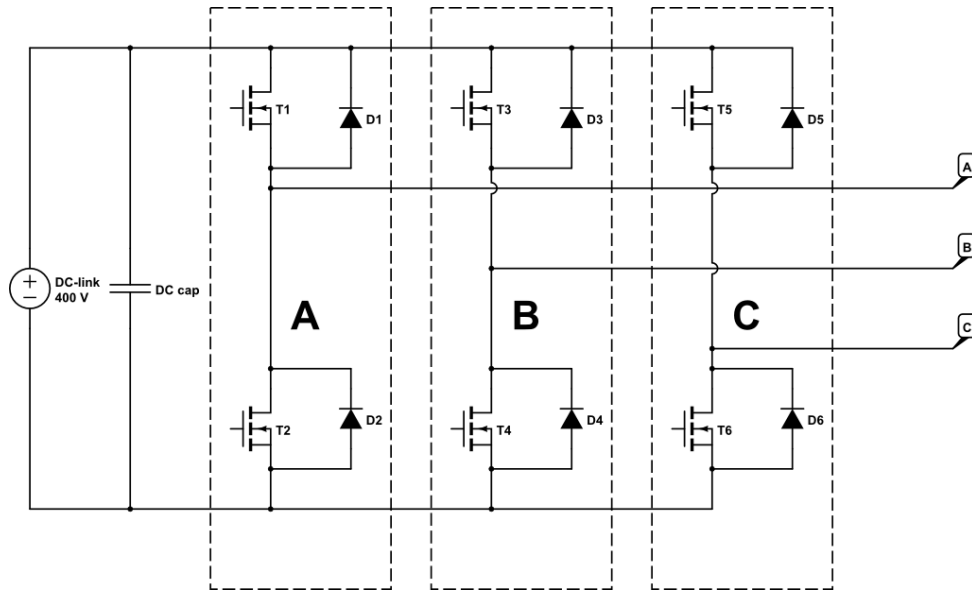


Figure 2.9: Three phase inverter without gate drivers

To convert DC to AC the inverter used in this thesis uses a technique called space vector modulation, by pairing the IGBTs in phase legs, A, B and C, three half-bridges is constructed with four possible states for each half bridge as can be seen in Figure 2.9. State one, the upper IGBT is on and the lower is off, then the positive DC-link is connected to the output of that phase leg. State zero, the lower IGBT is on and the upper is off, then the negative DC-link voltage is connected to the phase legs output. The third state can be referred to as passive zero, that is when both IGBTs are off and the load is only connected to the DC-link through one of the diodes in case there is an inductive current in the phase. The last state is a forbidden state and that is when both IGBTs are on, which will short-circuit the DC-link. The numbers following the state letter S in Figure 2.10 is which state each phase leg is in. For example S1(100) means that the upper IGBT is on for phase A while the lower IGBTs for phase B and C are on. By combining these states there is eight possible combinations if not counting passive zero and the forbidden states, the different combinations of states can be shown as vectors as can be seen in Figure 2.10.

During normal operation the controller to the PMSM calculates how the vector should jump between the different states to create the needed resulting rotating vector for the request. The rotating vector thereby creates an AC voltage which is supplied to the PMSM. If suddenly all gate drivers simultaneously are set in passive zero, the DC-link is only connected to the PMSM via the diodes. Where the inductive currents will go depends on the state they were in just before the gate-drivers were set in passive zero. The abrupt turn off will create a disturbance at that moment and if that disturbance is caused 1000 times a second the idea is that the resulting sound would be 1000 Hz.

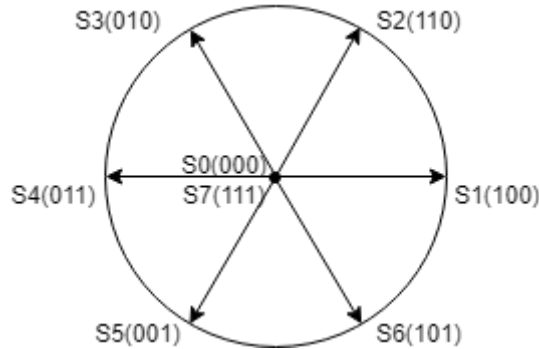


Figure 2.10: Switching pattern for first sector

2.4.2 Electric machines

An electric machine is a device that converts electrical energy into mechanical energy. It does so using electromagnetic forces. All electric machines have a stator and a rotor. How torque is produced differs between AC and DC machines as well as between different type of machines within both categories.

One type of AC machine is the synchronous machine, the synchronous machine has a magnetic stator field that is rotating at the same frequency as the stator current. Which results in that the rotor align itself with the stator field and the rotor will rotate synchronous, which is described in detail below. The number of pole pairs will together with the frequency of the current determine the steady-state mechanical speed of the rotor in revolutions per minute (RPM) by the formula;

$$N_s = 60 \cdot \frac{f}{n_p} \quad (2.6)$$

where f is the frequency of the current and n_p is the number of pole pairs. Since the amount of pole pairs is decided in the hardware design stage the only way of controlling the synchronous speed of the machine is by controlling the frequency. In modern power electronic applications, the frequency is controllable via an inverter. In the automotive industry the three-phase PMSM is common as a traction machine. The PMSM has permanent magnets in the rotor to create the magnetic flux. The design can vary when it comes to magnet placement, they can be glued on top of the rotor, creating what is commonly referred to as a surface mounted rotor topology. The advantage of surface mounting is that it results in a simpler manufacturing but not reliable due to the mechanical stress that the magnets are subject to under high speeds. They can also be inset mounted, meaning that it is grooved in the rotor, and the magnets are placed in the tracks. Compared to the surface mounted gives the machine better mechanical robustness. The third way is to place the magnets inside of the rotor and in the automotive industry, the magnets are often placed there to achieve a very high mechanical robustness [13].

Due to the magnets in the rotor, the rotor will always create a magnetic field and when a current is running through each stator winding, three new magnetic fields will arise according to (2.2). Since the current in the stator windings is of AC characteristics, the magnitude of each phase current will vary over time, in Figure 2.11 represented as the red green and blue sinusoidal curves. That would create an time varying magnetic field strength around the stator windings, where the magnetic field strength would be proportionate to the magnitude of the current. Each stator winding can therefore be either north or south pole depending on if the phase current is in its positive or negative value at every given moment. Since the stator windings are uniformly distributed in space, the point or points, depending on a number of pole pairs, where the magnetic field strength from the stator is strongest will move with the frequency of the current. Due to the natural behaviour of magnets the magnetic field from the rotor wants to align itself with the now rotating magnetic field of the stator. The arrow in Figure 2.11 shows how the rotor follows the magnitude of the magnetic flux, which are the red green and blue sinusodial curves. When the curves are at peak, the arrow points at that winding since it is where the magnetic field is strongest.

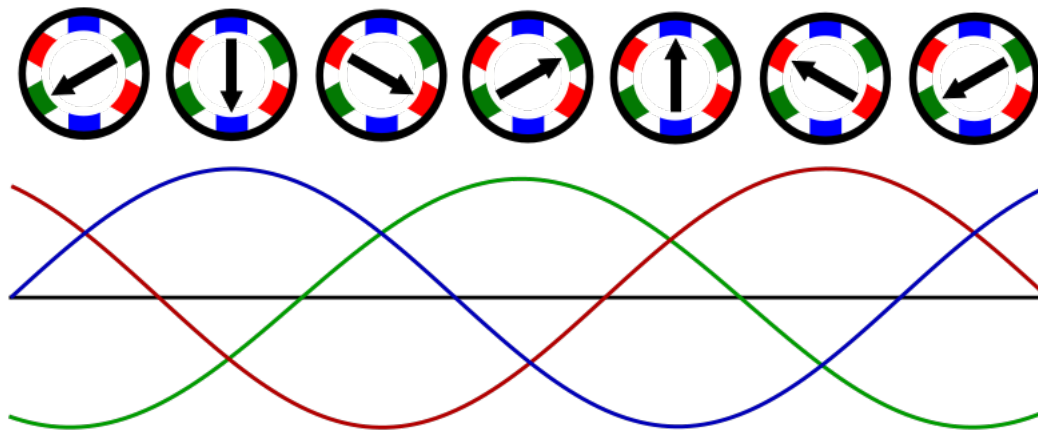


Figure 2.11: The rotor of an electric machines follows the magnitude of the current

2.4.3 PMSM circuit model

To easier foresee the electrical machine behavior the electrical machine can preferably be broken down into equations that describes the behavior. These equations can then be translated into an electrical circuit model for graphical representation. The simplest circuit model of a PMSM is when the machine stands still. One winding in an electrical machine is one long wire wounded several times inside the machine and forms a coil, hence the simplest representation of the PMSM is the RL-circuit shown in Figure 2.12:

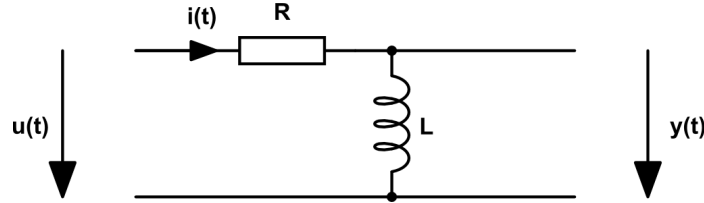


Figure 2.12: RL-circuit

Where the resistor, R , represents the resistance in the wire and the coil, L , is the inductance the wire creates when wounded. The expression for the voltage, u , and the current, i , is:

$$u(t) = i(t) \cdot R + L \frac{di(t)}{dt} \quad (2.7)$$

As mentioned, applying an AC-current to the three phase windings under the condition that the currents are phase shifted in respect to each other and that the windings are distributed uniformly, the stator windings will create a rotating magnetic field in the air gap of the PMSM. To optimize the use of the stator current, the position of the rotor is measured or estimated depending on what type of controller that is used and the position is fed back to the controller to ensure that the field moves with the rotor. This is done so that the rotor has the possibility to follow the stator field. When the machine starts to rotate, there will be now a rotating magnetic field from both the rotor and the stator. According to (2.3), a moving magnetic field induces a voltage. Thus, an EMF that opposes the input voltage is created, commonly named back-EMF. This phenomenon will be modeled as an AC-voltage source, as can be seen in Figure 2.13.

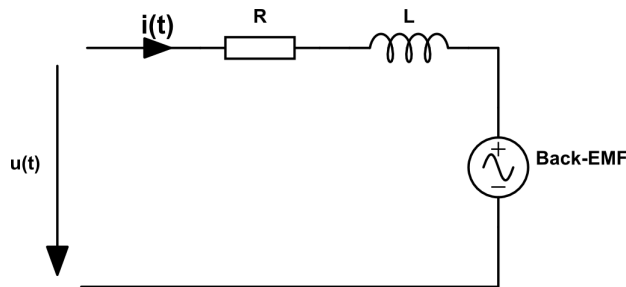


Figure 2.13: PMSM-circuit for one phase when the rotor is rotating

2.4.4 Dynamic modeling of the PMSM

For a PMSM to be used as a traction machine it needs some kind of controller. Because of its simplicity, the PI-controller is normally used for PMSMs. The base for the control strategy is a two-axis machine model [18]. However, all voltages and currents are AC signals and PI-controller are not designed to handle ever-changing values. It is therefore necessary to transform the AC quantities into DC quantities. The two-axis machine model will therefore be moved into a rotating reference frame to convert the AC signals to DC signals, the rotating reference frame is referred to as the DQ-plane.

The derivation from three-phase quantities to DQ is made considering the following assumptions:

- No iron losses
- Resistance is not frequency or temperature-dependent
- No zero-sequence component in the three-phase quantities
- The stator windings are sinusoidally distributed
- Linear magnetization characteristics

Start with the terminal voltages:

$$\begin{aligned} u_a(t) &= V \cos(\omega t + \phi) \\ u_b(t) &= V \cos(\omega t + \phi - \frac{2\pi}{3}) \\ u_c(t) &= V \cos(\omega t + \phi - \frac{4\pi}{3}) \end{aligned}$$

Transforming from a three-phase system to a two-phase system, also known as Clarke transformation [15], with amplitude invariant transformation can be done as:

$$\begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} u_a(t) \\ u_b(t) \\ u_c(t) \end{bmatrix}$$

The alpha-beta vectors can be represented in the complex plane where alpha is on the real axis and beta is on the imaginary axis. By letting alpha and beta components vary over the two axes there is one resulting rotating voltage vector, \underline{u}^s formed from the three-phase voltages, that can be seen in Figure 2.14 (a). Three-phase have been converted to two-phase but the quantities are still AC. Therefore, it is necessary to introduce a rotating coordinate system, which rotates with the same speed as the rotating voltage vector \underline{u}^s . This is called Park transformation [16] and the resulting coordinate system, commonly referred to as the DQ-plane, is widely used in these types of applications. To go from alpha-beta to DQ-plane

$$\begin{bmatrix} u_D \\ u_Q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} \quad (2.8)$$

Equation (2.8) is the Park transformation. θ must be an angle that changes at the same rate as the angular frequency of the rotor. It is common to pick theta as the angle in which the north rotor flux is oriented in relation to phase a, by doing so the d-axis for the rotating coordinate system would be the direction of the rotor flux, as can be seen in Figure 2.14 (b).

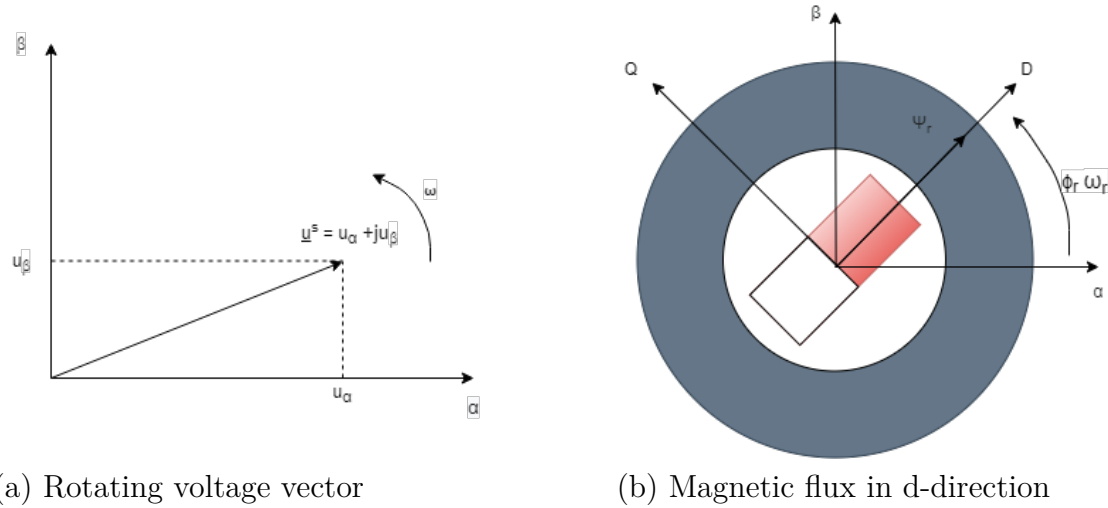


Figure 2.14: Alpha-beta to DQ system

To model the PMSM in the DQ-plane all three phase quantities from the PMSM needs to be transformed into a DQ-system and the model that can be seen in Figure 2.15 is used as base for this transformation.

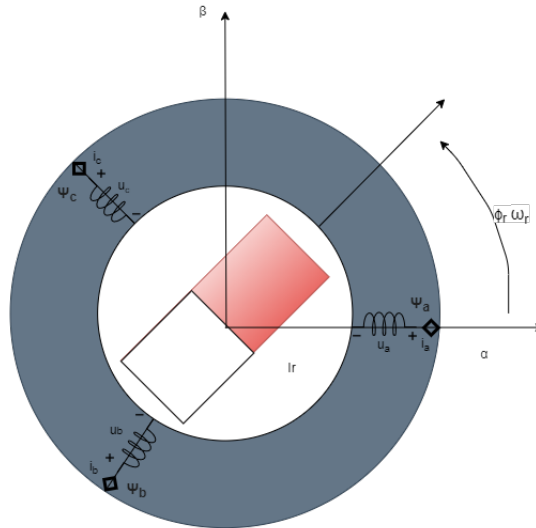


Figure 2.15: PMSM three-phase equations

From the model, the applied voltage to each coil is:

$$u_x(t) = i_x(t) \cdot R_x + \frac{d\Psi_x}{dt} \quad (2.9)$$

Where x is a , b or c depending on which phase that is modeled. As can be noted the difference between the PMSM and the RL-circuit presented in Section 2.4.3 is that the flux linkage term has replaced the induction term. According to (2.5) the total flux linkage in one phase will be a combination from all sources that are producing a magnetic field. The flux-linkage for one phase can therefore be described by the inductance of the coil itself, from here referred to as the self inductance, multiplied with the current in that phase. All mutual inductance's multiplied with each phase respective current as well as the flux linkage from the permanent magnets, multiplied with the angle of the rotor position ϕ since the amount of rotor flux reaching each phase varies with the rotor position:

$$\psi_a = l_{aa}i_a + l_{ab}i_b + l_{ac}i_c + \Psi_m \cos(\phi) \quad (2.10)$$

$$\psi_b = l_{ba}i_a + l_{bb}i_b + l_{bc}i_c + \Psi_m \cos(\phi - 120^\circ) \quad (2.11)$$

$$\psi_c = l_{ca}i_a + l_{cb}i_b + l_{cc}i_c + \Psi_m \cos(\phi + 120^\circ) \quad (2.12)$$

Where Ψ_m is the flux linkage from the permanent magnets, l_{aa}, l_{bb}, l_{cc} is the self inductance of each winding respectively and $l_{ab}, l_{ac}, l_{ba}, l_{bc}, l_{cb}$ the mutual inductance's between the different windings. If the rotor is designed to have saliency, which is common in traction machines, the machine inductance will vary with the rotor position. Meaning that some of the inductance in the machine varies with the position of the rotor. In that case, the inductance can be modeled to consist of two parts where the first part L_{aa0} represents the average value. The second part L_{aa2} is rotor position dependent. Note that magnets have equally strong flux density from both the south and north pole, which means that the rotor angle needs to be multiplied with two, as the north and south pole will have an equal impact on the inductance. The self inductance's of the machine can then be modeled as:

$$\begin{aligned} l_{aa} &= L_{aa0} + L_{aa2} \cdot \cos(2\phi_r) \\ l_{bb} &= L_{aa0} + L_{aa2} \cdot \cos(2(\phi_r - 120^\circ)) \\ l_{cc} &= L_{aa0} + L_{aa2} \cdot \cos(2(\phi_r + 120^\circ)) \end{aligned} \quad (2.13)$$

Since self-inductance has a rotor dependency the mutual inductance that each stator winding creates will therefore also have a rotor position dependency. Instead of having a peak value when the magnet is aligned to the coil the mutual inductance will be the peak value when the magnet is in between the windings. According to Gauss law [6], magnetic flux in any closed surface area always makes a closed loop. The flux that comes from one source and flows through the windings in one direction would be positive and on the way back that flux would result in a negative flux in other coils. Due to linear magnetization characteristics assumption, one simplification can be done, $l_{ab} = l_{ba}$ reducing the number of equations by half. This gives that the mutual inductance's can be described as:

$$\begin{aligned} l_{ab} &= l_{ba} = -(L_{ab0} + L_{aa2} \cdot \cos(2\phi_r + 30^\circ)) \\ l_{bc} &= l_{cb} = -(L_{ab0} + L_{aa2} \cdot \cos(2\phi_r - 90^\circ)) \\ l_{ca} &= l_{ac} = -(L_{ab0} + L_{aa2} \cdot \cos(2\phi_r + 150^\circ)) \end{aligned} \quad (2.14)$$

When (2.13) and (2.14) is inserted in (2.10), (2.11) and (2.12) respectively and the result of that is simplified. As a result, new terms will appear in the equations in multiple places. To further simplify, these terms is replaced by:

$$L_{sd} = L_{aa0} + L_{ab0} + \frac{3}{2}L_{aa2} \quad (2.15)$$

$$L_{sq} = L_{aa0} + L_{ab0} - \frac{3}{2}L_{aa2} \quad (2.16)$$

The final expressions for the machine in the DQ-plane is then:

$$u_{sd} = R_s i_{sd} + L_{sd} \frac{di_{sd}}{dt} - \omega_r L_{sq} i_{sq} \quad (2.17)$$

$$u_{sq} = R_s i_{sq} + L_{sq} \frac{di_{sq}}{dt} + \omega_r (L_{sd} i_{sd} + \Psi_m) \quad (2.18)$$

Where u_{sd} and u_{sq} are the d-q axes voltages, i_{sd} and i_{sq} are the d-q axes currents, L_{sd} and L_{sq} are the d-q axes inductances and ω_r is the electrical rotational speed. By translating the equations in circuits, the result can be seen in Figure 2.16.

Since one of the objectives of the thesis it so not have an impact on the average torque, it is therefore also of interest to see how the machine torque equation is described in the DQ-plane:

$$T_e = \frac{3n_p}{2} \left(\Psi_m + (L_{sd} - L_{sq})i_{sd} \right) i_{sq} \quad (2.19)$$

Where T_e is the electrodynamical torque and n_p is the number of pole pairs

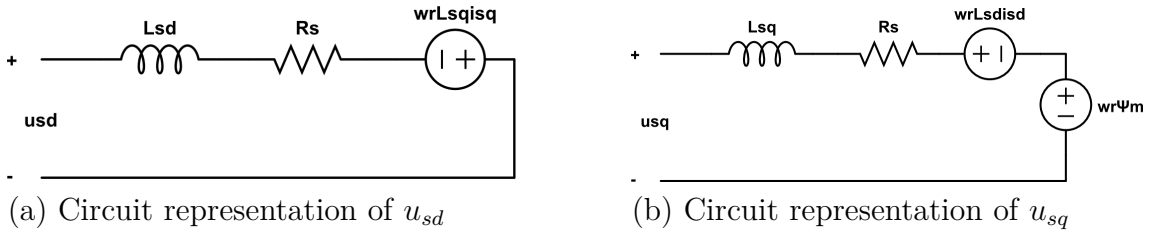


Figure 2.16: circuit representation of u_{sd} and u_{sq}

2.5 Electric machine controller

To be able to control the speed and torque of the electric machine a controller is needed. The block diagram in Figure 2.17 is a typical model over a control system for a PMSM connected to an inverter. The rotor position is measured by a sensor and then returned to the control system for the DQ-transformation to get the correct angle. The current from the inverter is measured and then compared to the reference value. This is called a closed-loop controller where measurements of the controllable quantities are utilized inside the controller. This is to ensure that the requested value is reached. The controller type used in this thesis is a PI-controller, the PI stands for Proportionate and Integral, which is how the controller reduces the difference between requested value and measured value. The controller receives the requested value from the user and if the sensor's reading is not matching the requested value there is a difference between the actual value and requested value. The proportional part of the controller then adjusts the output value proportionate to the size of the error and the integration part integrates the error over time and adjust the output more slowly. The value from the proportionate part and the integration part are added together and are converted into duty cycle which are sent to the gate drivers which controls the IGBTs with that duty cycle. This controller type should not leave a steady state error since that error is always reduced by the integrator part. However, there are limitations that could prevent this. For example, if the controller tries to achieve a value beyond the systems capability.

These controllers are usually designed as a first order low-pass filter because of that the step response have no overshoot when trying to reach the requested value. How fast a first order system reacts to a change depends on the systems bandwidth. The relation between rise time from 10 to 90% of the requested value and the bandwidth α_c is described by:

$$\tau_R = \frac{\ln(9)}{\alpha_c} \quad (2.20)$$

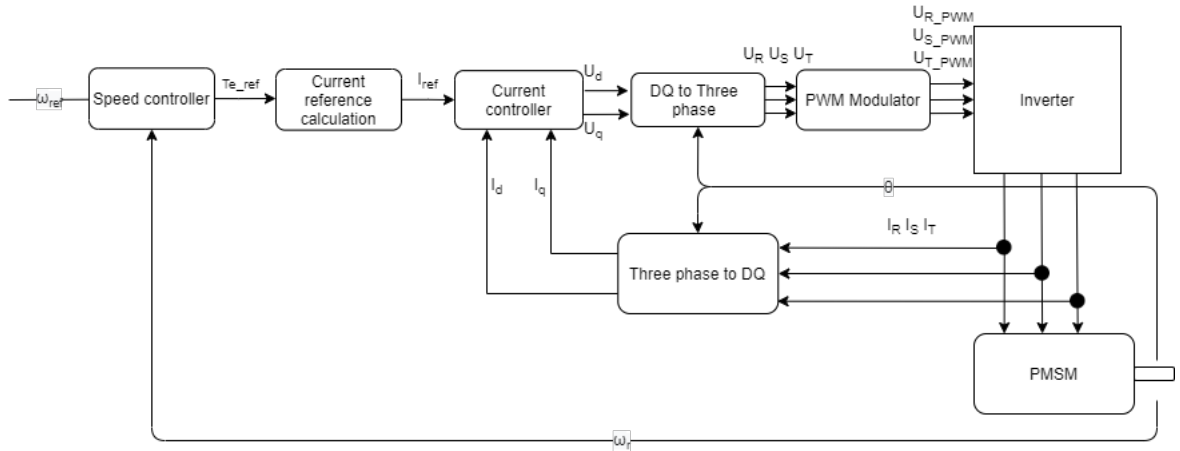


Figure 2.17: General model over the controller used

2.6 Implementation of the different methods

As mentioned before, to play music with an electric machine one can take advantage of the magnetostriction effect. One method to do this is to superimpose the music onto the voltage that is fed to the machine. This would give a variation in the flux linkage in the machine according to the superimposed music. In this case, the machine is controlled via a controller which means that the superimposition must be integrated as a part of the controller. This must be done with the requirement to not impact the average torque. If a signal with an average value of zero is used in the DQ frame, this will also have a net torque impact of zero in accordance with the scope of the thesis. A sinusoidal wave has these properties and therefore it is safe to superimpose a sine wave on the voltage that is supplied to the machine.

In the machine equations in DQ domain, described by (2.17) and (2.18) there is no sinusoidal term. In addition of that, if the bandwidth of the controller is low enough the AC-quantities will pass through without the controller noticing as the controller is implemented in the DQ domain which ideally only contains DC quantities. One can see the controller as a high-pass filter for such signals. Which means that signals that are changing rapidly is let through while slow changing signals are dealt with by the controller.

The simplest way to do the superimposition is to superimpose on either D or Q, to do it on both is unnecessarily complicated. Therefore one of D or Q must be chosen, if the wave that is applied is distorted, the average value may no longer be zero and if the music is supposed to be played when the machine stands still i.e. $\omega_r = 0$, one can see in both (2.17) and (2.18) that the cross-coupling term disappears. Which means that if the voltage is applied on the D-voltage in the DQ-system it wont result in any Q-current. According to (2.19), if i_q is zero there is no torque production regardless of the value of i_d . That is why it would be favorable to superimpose the signal on the D-voltage as can be seen in Figure 2.18.

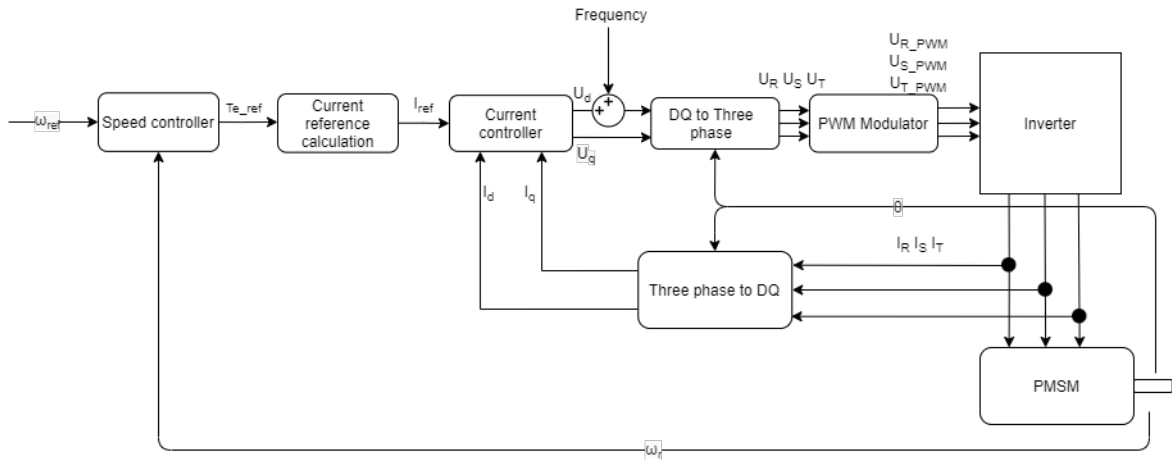


Figure 2.18: The superimposed signal are applied to the D-voltage inside the controller

Another method to play a song is by under impose a signal by setting all IGBTs in a passive zero mode which is described in Section 2.4.1. Even this method needs to be implemented within the controller and is done by sending a signal to the PWM-modulator block which disable the gate drives, i.e. setting them in passive zero. Where the signal is implemented in the controller can be seen in Figure 2.19. The result from the passive zero mode is that the PMSM do not get the voltage required, causing a temporary dip in the current. When this happens repeatedly it results in some current ripple at the desired frequency. The current ripple creates a magnetic field due to (2.2) causing magnetostriction which creates an audible sound.

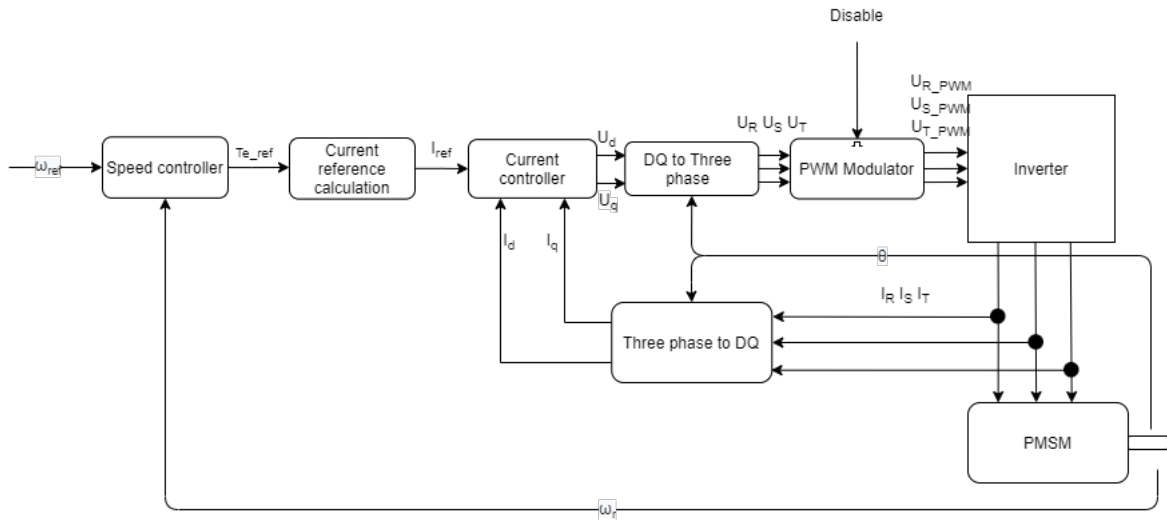


Figure 2.19: The stop-switching method is implemented in the PWM-modulator block to set gate drivers in passive zero

3

How to generate sound with an electric powertrain

This chapter describes the methods and the workflow of how the different methods were tested and evaluated. The evaluation will be done in two test rigs, one low power, the lab rig, and one high power, the production rig.

3.1 Test rigs

The methods were first tried out on a lab-rig that consists of two 12 V PMSMs mechanically connected back to back where one PMSM is in motor mode and the other in generator mode. The rig also has one inverter for each motor, the inverter can output DC-voltages up to 500 V and drive loads up to 150 kW. The inverter comes with built in current sensors which can measure currents up to 700 A peak. The resolution of the current sensors are a minimum of 0.4 A, which means that anything less than that cannot be detected. The inverter is controllable via two different software and the lab rig can be seen in Figure 3.1. The lab rig is designed in such way that there is a one to one mapping to the production rig, which means, in theory that if one software runs on the lab-rig it will also run in the production-rig. With one exception, the lab-rig is made for experiment and therefore all code to the lab-inverter is accessible in another way than the code for the production rigs inverter. This results in that only the superimpose method will be able to run on the production-rig as it is during the time of this thesis work as the part of the software that configures the gatedrives are not accessable. Aside from an inverter, the production-rig have a PMSM of the same sort as are mounted in Volvos electric vehicles. That PMSM is mounted to a dynamometer instead of the another PMSM for better control over the load torque.

All models discussed are built in SIMULINK which is a graphical based programming language made by Mathworks. Constants and vectors that are input to some of the blocks in the SIMULINK models are mainly produced in MATLAB scripts. The SIMULINK models are compiled and then flashed onto the processor, the processor itself is rather fast with a 200 MHz clock. But the execution window of the software is 100 μs which limits the number of frequencies playable in different ways with the different solutions. Also, memory is an issue for one solution, which also is discussed further in the next section.

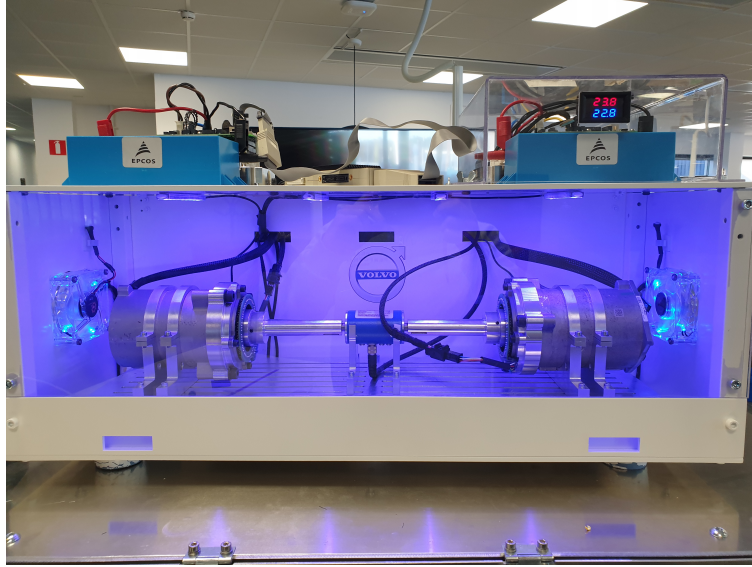


Figure 3.1: The small test rig

The software used to request user inputs to the inverter which by extension also the PMSM is CANalyzer. CANalyzer is a software where CAN signals can be monitored and altered. For monitor the code running in the inverter, a JTAG debugger by Lauterbach is used. JTAG is a standardized connector for testing and verifying printed circuit boards designs after manufacturing [21]. The debugger is connected to the inverter via JTAG and comes with the software Trace32, in which it is possible to track variables in real time. In addition, a software from ETAS called INCA were used to extract internal measurement points used inside the controller. Some of these measurements are presented in Chapter 4.

3.2 Superimpose

As mentioned in Section 2.6, the bandwidth of the controller determines the lowest frequency that can be superimposed. The lowest frequency of the sine component can be determined by the time period described by (2.1) and the rise time of the step response determined by (2.20). In the lab-rig case, where the bandwidth was set to five, the rise time is:

$$\frac{\ln(9)}{5} = 0.44s \quad (3.1)$$

To have five in bandwidth is very low and the value were chosen because the current measurement units in the lab-rig are designed to measure much larger currents and therefore there were much measurement noise present that the controlled tried to act on. This caused the motor to oscillate fast and were hard to control if the bandwidth was selected too high. By lowering the bandwidth of the controller, it became much slower and did not act as much on the noise as before.

A sine component of 100 Hz or more which have a time period that is 4 times as fast as the rise time, could therefore follow through the DQ to three-phase transformation and be superimposed on the three-phase voltage to the machine without

impacting the controller and therefore also the average torque.

The frequency that is superimposed is in the range of 0.1-3.3 kHz. The lower limit of 100 Hz is both a physical and a practical limitation, it is possible to produce low frequencies with an electrical machine but it won't result in the same bass effect as a sub-woofer would create since it is not possible to move large masses of air. In addition to that is that low frequencies would not pass through the controller. Since the low frequencies on the voltage will result in a low frequency current according to ohms law, which would be within the bandwidth of the current controller and thus be suppressed by it. However if there is a frequency that is desirable to play, it is possible to temporally lower the bandwidth of the controller. By doing so, that would risk that the controller might not be able to produce the amount of torque that is requested and are therefore is not covered in this thesis.

The upper limit of 3.3 kHz is a limitation of the inverter, since the switching frequency of the inverter is set to 10 kHz. It would be possible to create frequencies over 3.3 kHz but to minimize the risk of aliasing, the highest value were chosen to have some margin to the maximum value.

Another issue with this method is hitting the voltage limit, which results in that the signal no longer would be a sinusoidal wave and it's hard to predict what kind of signal that will be generated. This can be avoided by setting a low amplitude on the sinusoidal signal on the superimposed voltage since it's a one to one mapping in amplitude, due to the earlier mentioned reason that there is no sinusoidal dependency in either (2.17) or (2.18), the amplitude of the superimposed signal will therefore be the same in the three-phase quantities.

3.2.1 Tone creation for superimpose method

To superimpose a tone on the D-voltage in the controller, some signal must first be created. The tone could be created directly within SIMULINK or it could be imported to the SIMULINK. Both of these ways need to consider how often the code is executed. CPUs execute instructions with a certain clock speed, that clock speed could be used to create a counter within a program with the same resolution of the time step. The resolution of the counter used here would then be $100 \mu s$, since the code is executed at the switching frequency which is 10 kHz. The output from the counter designed here is an increasing value that updates every iteration until some threshold is reached, then the counter resets. To use the counter to create a tone the output from the counter must be multiplied with a sinusoidal function as:

$$y(t) = A \cdot \sin(\omega t) \quad (3.2)$$

where A is the amplitude, $\omega = 2\pi f$, f is the frequency and t is the time from the counter. The relation between frequency and time period can be seen in (2.1). By choosing a reset value to the counter that corresponds to one full period for a certain frequency in a sinusoidal function i.e reset value = T . $f(t)$ would then be a sinusoidal function with the frequency f and amplitude A .

By implementing two of these counters with a frequency of 500 Hz and 1000 Hz respectively with a third counter that alternates between the two first counters every second the result can look like in Figure 3.2.

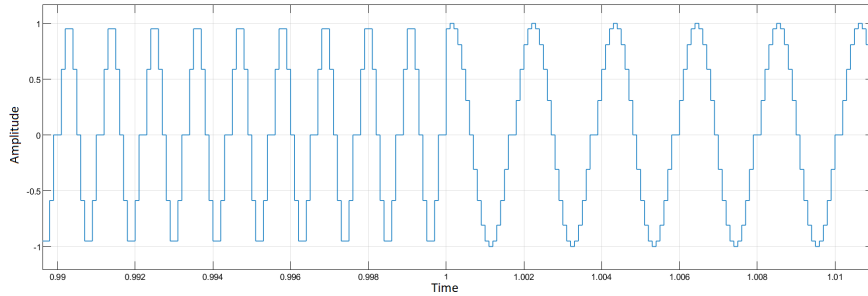


Figure 3.2: Output from different counters

To implement this solution to play music would need each of these counters multiplied with a sinusoidal function in order to create every unique frequency needed in a song. Counters would be needed to control in which order every frequency should be played, this would result in an incredibly complicated network. In addition to that, as can be seen in Figure 3.3 (b) songs do not always have pure sinusoidal waves but rather a combination of multiple frequencies superimposed on each other. To play music with the solution above would not be practically possible to do. Therefore some other solutions were needed.

The song "Train" from Karl Lindblom was imported into MATLAB with the audioread function. The audioread function is a inbuilt MATLAB function which are

able to read the most common audio file types. The audio data in the imported file is returned in MATLAB as an m-by-n matrix and also the sample rate, F_s which the audio was recorded with. The m is the number of audio samples and n is the number of channels in the file [22]. The result can be seen in Figure 3.3

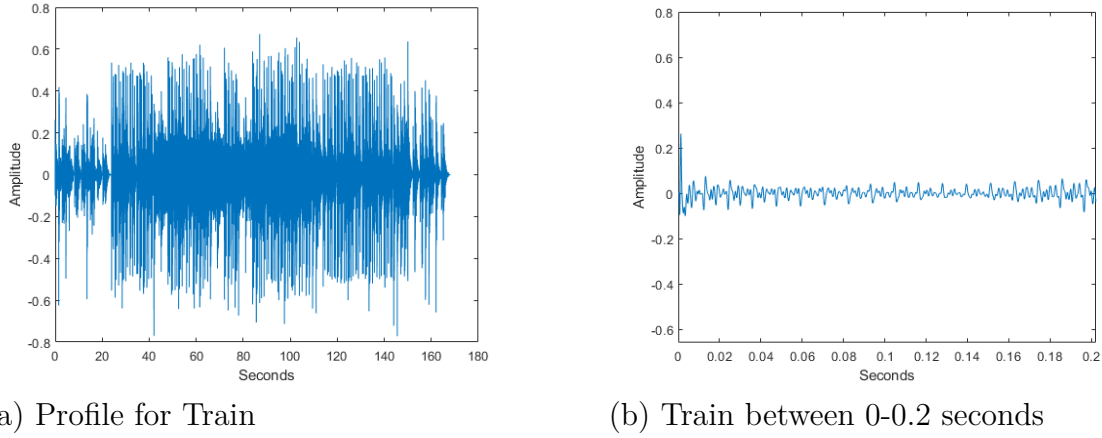


Figure 3.3: Train by Karl Lindblom

Figure 3.3 (a) would be the current profile that would be sent to a speaker. The Y-axis is the amplitude of the current and the x-axis is the value at each second, as can be seen in Figure 3.3 (b). The song is 168 seconds long and it is sampled with a frequency of 44100Hz which gives a file with $168 \cdot 44100 = 7408800$ numbers of samples in total. Because of memory limitations it's not possible to directly superimpose the entire song on the D-voltage. In addition to that the switching frequency is much lower than the rate the song is sampled with. Therefore, if one wants to play the entire song, the song needs to be downsampled. There are multiple ways of downsampling a signal, one easy method is to take a fixed amount of samples and take the average value of those, this works well if the number of samples is limited to a maximum of 10, otherwise, the song gets too distorted to be able to hear the original song and even if the song got downsampled by 10 times it was still too big to be able to import on the CPU because of the limited memory. A more advanced method is to use Fourier transforms. As described earlier, the Fourier transform breaks down a signal into frequency components. An algorithm was developed to take a preset number of samples of the imported song, made a Fourier transform on that number of samples, saved the most dominant frequency in that interval, and stored that frequency to a vector. Since samples are taken over time, a time vector is created simultaneously with a length that corresponds to the number of preset samples. The length of each time step can be calculated as follows:

$$\text{Time step} = \frac{\text{Amount of samples}}{\text{Sampling frequency}} \quad (3.3)$$

By taking a fixed sample amount of data that is equal to a quarter of the sampling frequency, one value each quarter of a second could be captured, which means that the song is reduced to $168 \cdot 4 = 672$ number of values, which is much fewer values than before and its no issue to store that amount values.

There is a major drawback with this method, if the amount of samples taken from a song is constant, as can be seen in the example shown in Figure 3.4 (a), section one and two happens to be sampled correctly but section three gets half the pause and half the tone which returns the most dominant frequency of the tone and pause. As a result, the pause section of the song is removed in the new down sampled vector. This causes a large difference between the down sampled song the original. This will likely distort the down sampled song so much that the perceived song when played on the PMSM is not the same as the original song.

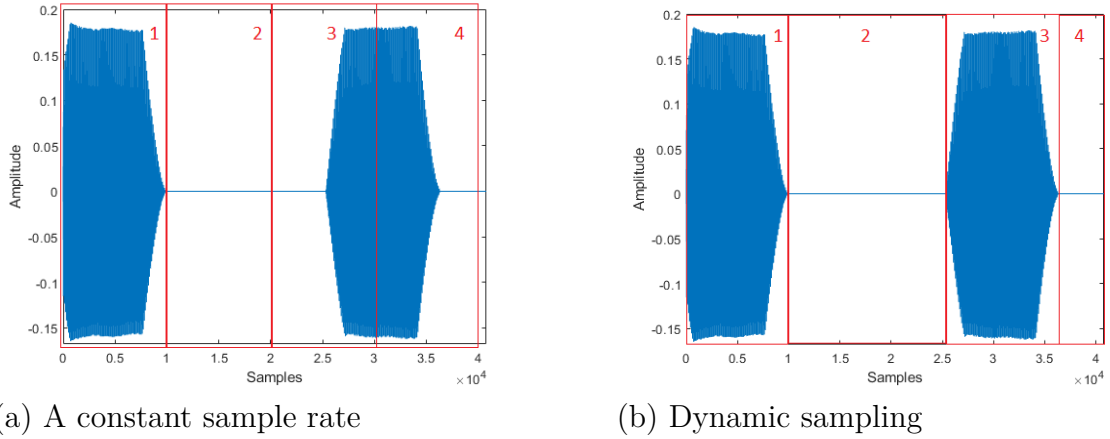


Figure 3.4: Difference between dynamic and constant sampling

A solution to this is to introduce dynamic sampling, instead of having a fixed number of samples, an algorithm tries to figure out if the samples it's looking at is a tone or a pause, by doing so, the most dominant frequency of each tone segment is captured as well as the pauses in-between. An example of dynamic sampling can be seen in Figure 3.4 (b) where the tone and pause is captured correctly and this results in a vector that better represents the original song. The code for both programs as a whole is presented in the appendix.

3.2.2 Implementation of superimpose solution

When the song has been down sampled and stored in a vector it needs to be implemented in the SIMULINK controller model. This solution were integrated in the current controller part of the complete controller, as can be seen in Figure 2.18. The current controller were chosen because the D-voltage is accessible in that part of the controller.

The vector containing the frequencies and the associated time vector are both stored in a lookup table for simpler indexing. How the values are stored can be seen in Table 3.1.

Counter A has a reset time equal to the length of the song, which means that the output from Counter A will increase from zero to the reset value in steps of the time period of the inverter. The output from Counter A is connected to both the

input of a look-up table, and the input of a sinusoidal function. When the time value of Counter A reaches a corresponding time value in the look-up table, the look-up table outputs the matching frequency. For example as can be seen in table 3.1 at time 0.35 the frequency 660 Hz will be outputted. At the same time there is always an increasing value on the time as argument, as can be seen by (3.2) those together will create a sinusoidal signal with the frequency of choice. The result is a sinusoidal function with the desired frequency, the resulting function is then added on the D-voltage. How the method is implemented in a graphical way can be seen in Figure 3.5.

Table 3.1: Frequency and time vectors as they are stored in the look-up table

Frequency	time
660	0.1
0	0.25
660	0.35
0	0.65
660	0.75
0	1.05
510	1.15
0	1.25
660	1.35

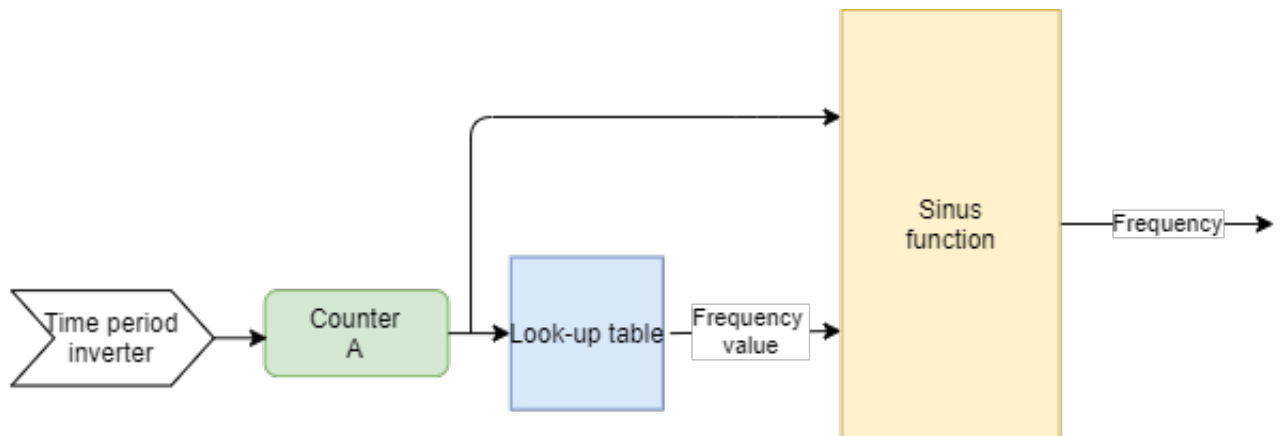


Figure 3.5: Implementation of the superimpose method as a block diagram

3.3 Stop-switching method

As mentioned the switching frequency is 10 kHz, thus, the frequency range in which sound is able to be generated is 0-10 kHz since the current controller can only update that fast. The major difference between this method and the superimpose method is that the sound creation does not happen because frequencies are added to the voltage. Instead it is the removal of time when the machines is supplied via the inverter that creates the sound. The solution was designed to be as simple as possible inside SIMULINK since it is the SIMULINK code that runs on the rig, the more efficient code the more likely it is that the processor have time to execute all instructions within the execution window. That leads to that frequencies have to be translated into time periods instead, the relation between time and frequency is described by (2.1). Note that this is an inverse relationship, which will, due to that the minimum step size that can be taken is $100\mu s$, because of the execution time that is 10 kHz. The result of that is that it creates discrete steps of frequencies where the resolution among the high frequencies are low since to go from $1/100\mu s$ to $1/200\mu s$ is an increase of 100%. As a result the resolution among the low frequencies becomes higher i.e. To go from $1/1400\mu s$ to $1/1500\mu s$ is only an increase of 7%. Resolution examples can be seen in Table 3.2

Table 3.2: Resolution becomes higher among lower frequencies due to the inverse relationship between time and frequency

Time step [μs]	Resulting time period [μs]	Frequency [Hz]
100	1/100	10000
200	1/200	5000
300	1/300	3333
...
5000	1/5000	200
5100	1/5100	196
5200	1/5200	192

The effect of this can be that a song cannot be reproduced perfectly and some alterations in the original song have to be done. This solution does not need to add a sinusoidal waveform on a voltage and therefore it gets easier to implement code-wise.

One downside with this method is that the abrupt turnoff does not only create the harmonic that is wanted but also harmonics that are hard to predict and control. When using the stop-switching technique a small modification to the PI-controller was done to mitigate the absence of switching. The controller strives to meet the requested current by changing the pulse width as described in Section 2.5. When suddenly the gate drivers are set in a passive zero mode some of these pulses are never sent. The controller wont be as responsive as it was before the stop switching method were implemented and to compensate for this is to modify the gain on the controller to be dynamic and to faster compensate for the disturbance the stop switching method introduces to the system. Since the song that is played is

known the information can be supplied to the part in the current controller where bandwidth is set and multiply the bandwidth with a corresponding gain, based on the number of passive zero states.

By making two vectors in MATLAB, one for time periods and one for corresponding gain the `cftool` in MATLAB could be used to get a function. The `cftool` is a curve fitting application within the Math, Statistics and Optimization package from mathworks [24]. The gain values were chosen to increase the gain with the percentage loss that the stop switching results in. i.e if the loss is 1% the corresponding gain is 1%. As previously mentioned the frequencies must be stored as the resulting time period each frequency have, which can be seen in Table 3.3.

Table 3.3: Time periods for chosen frequencies and corresponding desired gain value

Resulting time period	Corresponding gain
1/100	1.01
1/200	1.02
1/300	1.03
1/400	1.04
1/500	1.05
1/600	1.06
1/700	1.07
1/800	1.08
1/900	1.09
1/1000	1.1
1/2000	1.2
1/2500	1.75
1/5000	2

Since the gain increases exponentially the curve fit for a power function would be most suitable, for higher precision two terms were selected and then the Least absolute residuals (LAR) robustness since no extreme outliers were included in the vectors, the algorithm used were the pre-selected trust-region.

3.3.1 Implementation of the stop-switching solution

As mentioned the stop-switching method is also implemented in SIMULINK. The MATLAB-script that convert songs to frequency vectors were written to work for both methods which lead to that parts from the superimpose method were re-used for this method. The re-used parts were that the time period vector and corresponding time vector needs to be stored in a way which allows easy indexing, and that indexing needed counters. Instead of the current controller the stop-switching method is implemented in the PWM-modulator block which can be seen in Figure 2.19. The PWM-modulator block were chosen because all three gate driver signals could be intercepted there.

3. How to generate sound with an electric powertrain

One counter, in Figure 3.6 called Counter A, have a reset value equal to the length of the song in time. Counter A counts with the time period of the inverter i.e. steps of $100\ \mu s$. Counter A is the input to a look-up table containing the frequency periods vector and time vector. As in the superimpose case, when the time value of counter A reaches a corresponding time value in the look-up table, a resulting period is chosen and sent to Counter B as reset value.

Counter B also counts with the time period of the inverter. Counter B gets the output value from the look-up table which is the period value of a frequency as its reset value. Hence, Counter B is now resetting once each period of that frequency, each time it resets, a switch sends the signal to set the gate drivers in a passive zero state for one clock cycle. This is repeated with the frequency of choice to produce the desired sound.

For example if Counter A have reached one second, the stored value for one second in the look-up table is for example $1/660$ which is equal to 0.0015 , the look up table outputs 0.0015 to Counter B. Counter B will count with steps of 0.0001 until 0.0015 is reached, Counter B resets and by resetting the signal to set the gate drivers in passive zero is sent. Under one second this repeats 660 times, which results in a current ripple of 660 Hz.

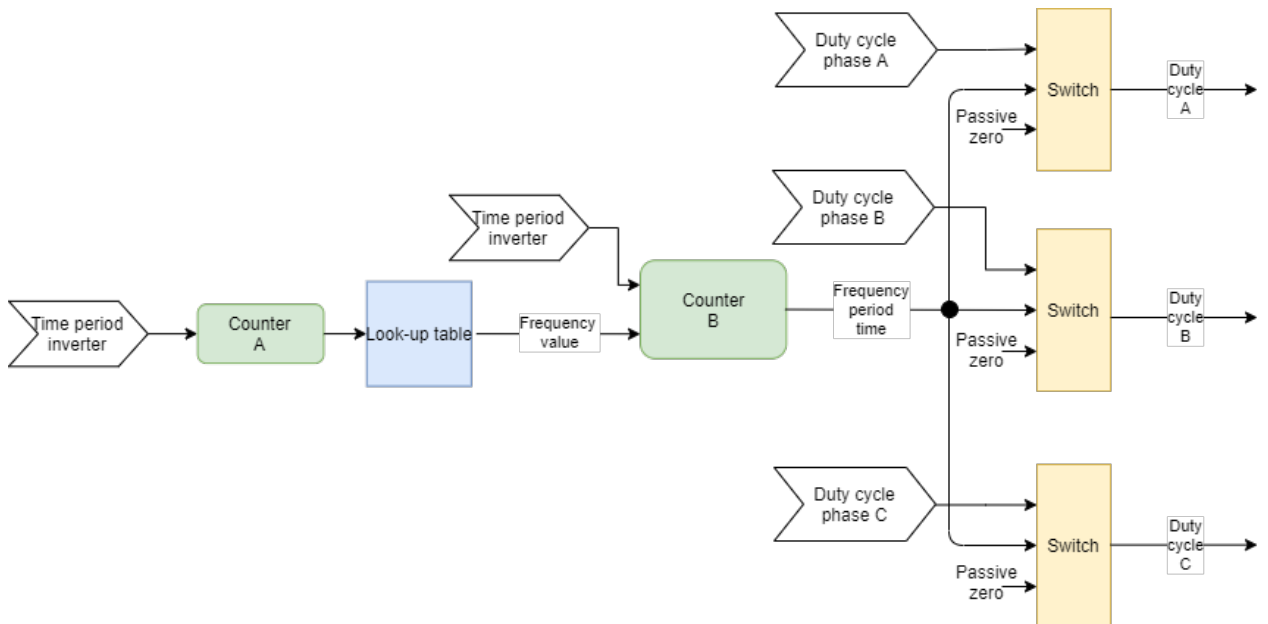


Figure 3.6: Implementation of the stop-switching method as a block diagram

3.4 Acoustic transfer function of the lab rig PMSM

In order to find the acoustic transfer function of the PMSM, both solutions were subject to the same evaluation. The reason behind to test both methods were to find out if there would be any difference in sound with the different methods. For both tests the same 16 frequencies were chosen to be sent to the lab rig PMSM. The frequencies were chosen to start from 100 Hz increase with 100 Hz. For frequencies over 1000Hz the stop switching method have low resolution and therefore the remaining frequencies were chosen such as they could be played with both methods. The counter value presented in Table 3.4 is what the look-up table outputs in order to produce the corresponding frequency. The response was then recorded and analyzed.

Table 3.4: Counter value and what frequency that results in

Counter value	Corresponding frequency
0.01	100
0.005	200
0.0033	303
0.0025	400
0.002	500
0.0016	625
0.0014	714
0.00125	800
0.00111	909
0.001	1000
0.0008	1250
0.0007	1428
0.0005	2000
0.0004	2500
0.0003	3333
0.0002	5000

The stop-switching method was the method that returned the clearest tones at the time when the transfer function were to be obtained and therefore that method was tested more comprehensive. The superimpose method was only tested with the final method that came from these tests. There was two different methods tested. The first method to obtain the transfer function was to place all frequencies in a vector and play each frequency for five seconds while recording the frequency response with a microphone. The recording were done with a MATLAB-script using the Record and Play Audio functions. The functions are in-built MATLAB functions which are able record sounds from a microphone connected to the computer and play the recorded sound on speakers [23]. In this case the frequency response from this method was recorded with a Sony MDR-XB950N1 headphones at a sample rate of 10 kHz. The frequency response was then analyzed with the same script that was used to analyze songs. An FFT was done on each step in order to analyze which

3. How to generate sound with an electric powertrain

frequency the system responded with. The script can be found in appendix.

To obtain the acoustic transfer function in one large measurement made it hard to analyze. Interference is always present and when doing one large measurement run, it is hard to differentiate between if the interference is a part of the acoustic transfer function or if it is just interference. Even though the test was done for a non-rotating case, the rotor would still make small movements sometimes due to measurement noise from the current measurement units. When the rotor moved slightly a clicking noise could arise or the tone would change characteristics a bit. As a result, to have more robust measurement data, a different method was tried. Each frequency was treated in separate measurement sessions one by one for a fifteen second duration each. When the tones were played one by one, the issue with a sound that could appear in the transition between sounds were eliminated.

The response from the second method was recorded with both the Sony MDR-XB950N1 headphones also at a sample rate of 10 kHz but also with a professional recording equipment called SQadriga. SQadriga is a mobile four channel recording equipment made especially for industry applications. With the SQadriga recorder comes headphones which also have microphones on each side. The headphones were placed inside the test rig box as can be seen in Figure 3.7. The recording with SQadriga was done in stereo, each frequency was played for fifteen seconds and sampled with 44100 Hz. The higher sampling rate eliminates the risks of aliasing mentioned earlier.

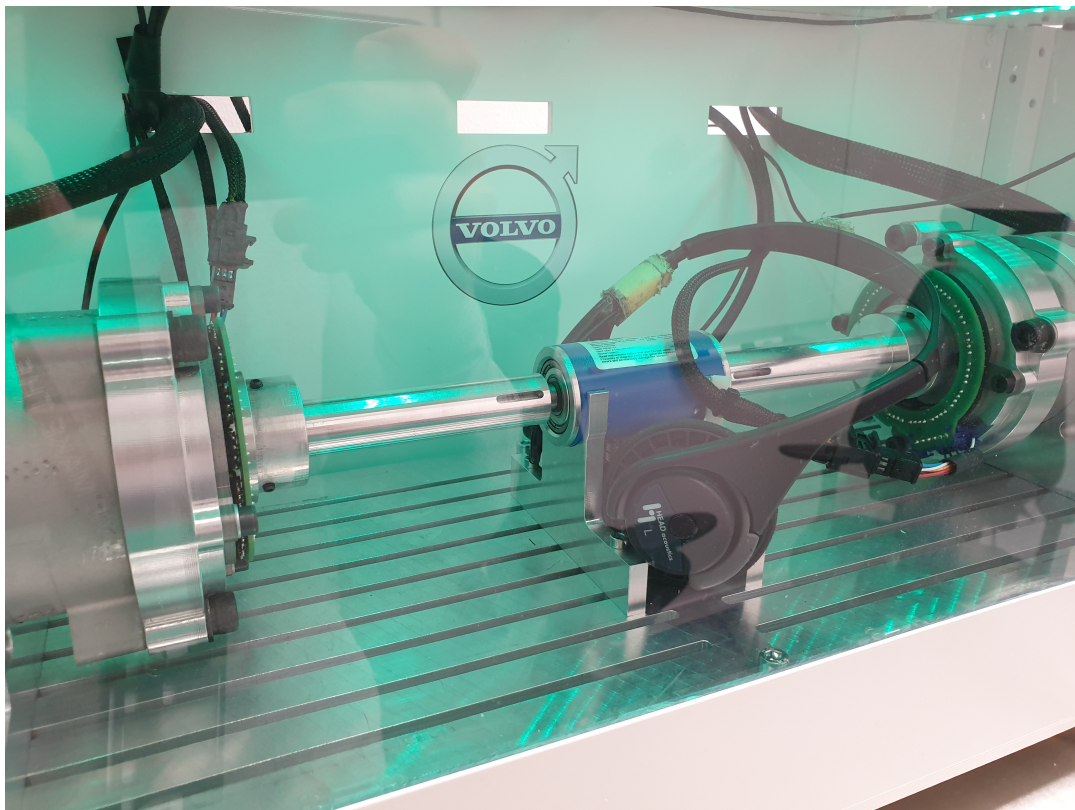


Figure 3.7: SQadriga Recording equipment inside rig

4

Measured sound from the test rigs

As stated, the aim of the thesis is to play music with an PMSM without affecting the average torque. The torque could not be measured directly in the test rig. However, the current could be measured and with (2.19) the torque can be calculated using the measure currents. If i_{sd} and i_{sq} is unaffected when music is added to the system, then the aim of the thesis is considered to be fulfilled. The reason why some deviation is allowed is that adding music to the system will introduce some disturbance which, even if it in theory should not, will affect the currents somewhat. In addition to that there is also constant measurement noise which needs to be considered as well as the resolution of the measurement devices. If the result is within that noise and measurement uncertainty then it is hard to differentiate between normal operation noise and the impact from playing music.

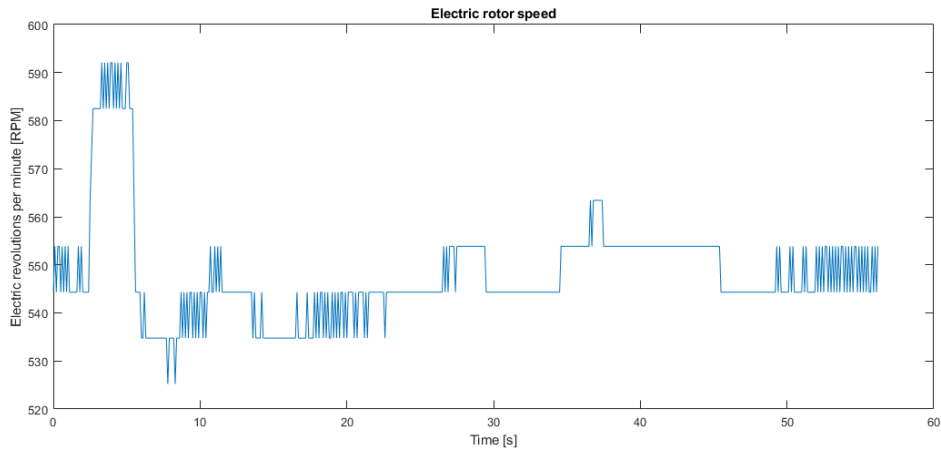
4.1 Measurements using the superimpose method

4.1.1 Results from lab rig

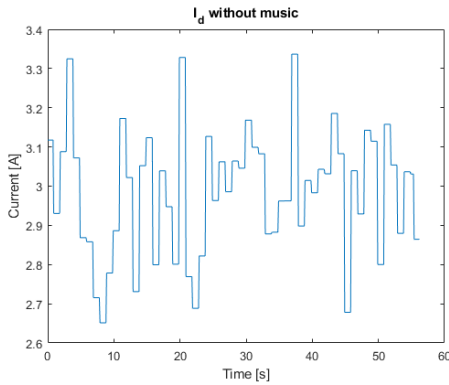
When the machine was at standstill and the test described in Section 3.2.1 with the alternating frequencies between 500-1000Hz was done, the PMSM returned an audible difference between the requested frequency and the sound the machine made. When playing a song, with the machine still in standstill it was clear that the tones changed but it was hard to hear which song that was played. Also some frequencies caused the rotor to rotate even without any DQ-current request. This issue became even worse if a higher amplitude was used on the superimposed voltage. That the rotor rotate even though there is no current request is an unwanted behaviour since the goal was to produce music without impacting the average torque. However, there is no external load on the rotor and its mechanical inertia is limited as this is a small machine. It is possible that if there was a load this would not be an issue. Because of the small inertia and the lack of a torque sensor it was difficult to estimate the amount of torque produced. To test the theory that it would not be an issue if there was a load, the other PMSM connected in the back to back setup was short circuited to create a load. The higher load removed the issue with the rotor starting to move when only the superimpose voltage was applied to the machine, which strengthen the theory that this would not be an issue if the PMSM was mounted in a car.

4. Measured sound from the test rigs

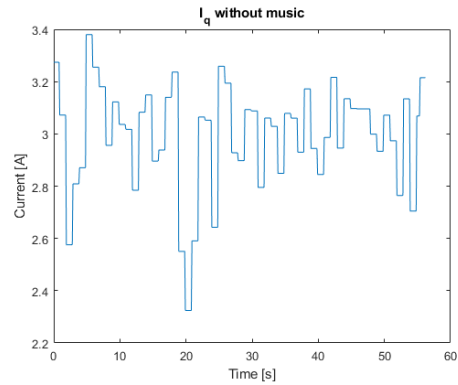
After the stand still test, a test when the machine was rotating was performed. For the non-standing still operation, the PMSM connected back to back were disconnected. Which means that there was no load connected. Three amperes were requested in D and Q current. In Figure 4.1 the results from this test when no music was added is shown. As can be noticed from the figure the speed is rather constant for this operation.



(a) Electrical rotor speed when no music is added to the system



(e) The average value of I_d when the value was calculated over ten samples. The average over the entire period is 3A



(f) The average value of I_q when the value was calculated over ten samples. The average over the entire period is 3A

Figure 4.1: Rotor speed and current measurements when a rotational case was performed and no music was added to the system.

When repeating the same test as in Figure 4.1, but superimposing a song on the requested voltage, the PMSM returned the same audible sounds as in standstill, which means that it was hard to identify which song it was. However the rotor speed was stable, when the amplitude of the superimposed voltage was low. In this case it was one volt. The speed from this test can be seen in Figure 4.2.

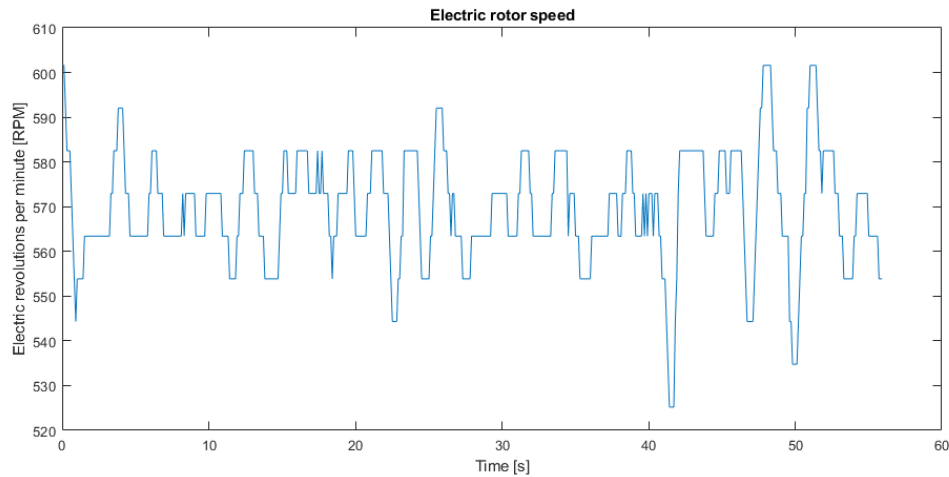


Figure 4.2: Electric rotor speed with superimposed song when 3A were requested in D and Q currents

The speed can be compared to the rotor speed when no song was superimposed, shown in Figure 4.1. The speed fluctuates a bit which is a result of measurement noise and the superimposed voltage. As one can observe the speed fluctuates around 570 electrical RPM in Figure 4.2 as can be compared to 550 electrical RPM in Figure 4.1. That the speed is higher can be an indication of that the average value of the superimposed voltage is higher than zero and that it is an resulting current that is added.

4. Measured sound from the test rigs

The oscillations from the superimposed voltage can be seen in the PWM pattern in Figure 4.3. The PWM width increases some when there is some noise on top of the graph as can be seen in the same figure, while as it decreases some when there is noise in the bottom of the graph. That can be compared to when there is a pause in the song, where the PWM width is the same as can be seen in Figure 4.4. That the PWM-width is changing when the song is played reveals that it is the superimposed voltage that is playing the music and that the method is working as intended.

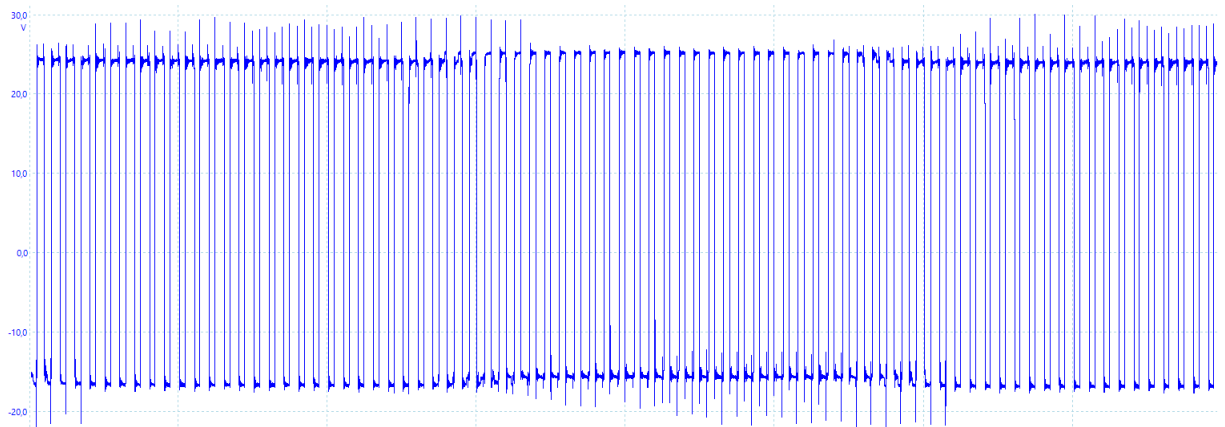


Figure 4.3: PWM with song

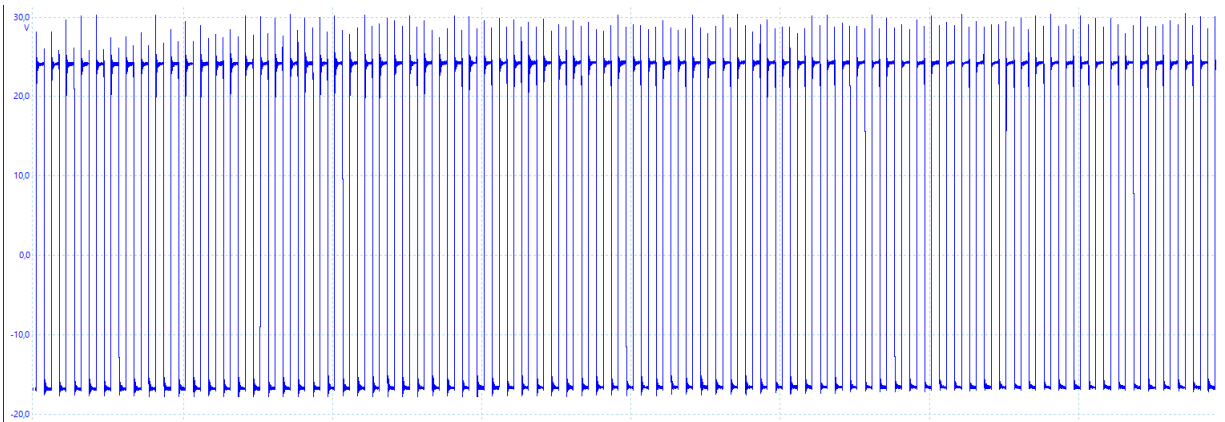


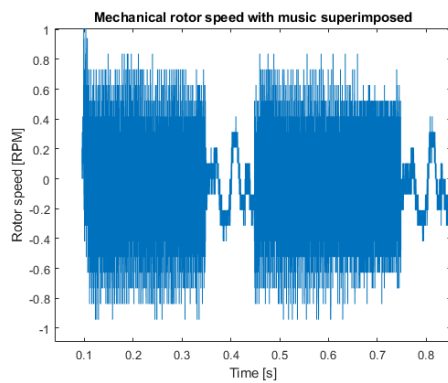
Figure 4.4: PWM-pattern from inverter to the machine when song is superimposed

4.1.2 Results from production rig

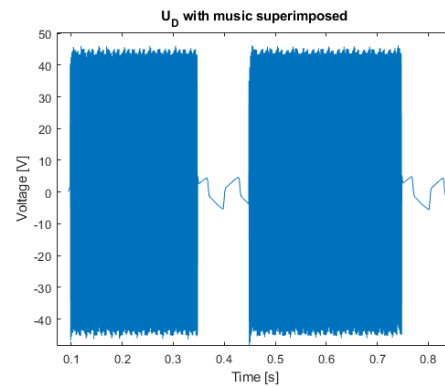
The only method that was tried on the production rig was the superimpose method. Due to limitations with the software that controls the gate drivers in the inverter. The signal that is used in the lab rig to set the gate drivers in a passive zero state is not accessible in the same way in the production rig as in the lab rig.

In the production rig, for safety reasons people are not allowed to be in the same room as the test objects. All control of the machine and inverter is handled by a control panel in an adjacent room, but there are a microphone installed in the room with the test object which are connected to speakers located by the control panel. In the room there is a big fan to cool the dynamometer. This fan is very loud which makes it impossible to hear any noise coming from the PMSM even if one would stand inside the room. When running a test the graphs indicated that the song was playing but it was impossible to confirm due to mentioned fan noises. Therefore an operator at the rig temporarily turned off the fan. However, turning off the fan came with restrictions, the dynamometer could not be used since it was no longer cooled. When the fan was turned off it was possible to hear the song that was played on the PMSM. In the author's opinion, superimposed signal sounded much better on the production rig compared to the lab-rig. Since the sound was recorded and played via a speaker the sound is already a bit distorted but the result sounded very similar to the stop-switching result. It was clear which song that was played. Which it wasn't when the method was tested on the lab-rig.

First the non-rotational case was performed, one drawback was that the superimpose signal leaked through a bit. From 0.1 second to 0.35 is a tone followed by a pause and then a tone again, as can be seen in figure 4.5 (b), which made the current controller try to regulate it as can be seen in 4.5 (a), there is much more noise when the signal is played than when it is not. This resulted in that the rotor could be heard struck back and forth during operation.



(a) Mechanical rotor speed

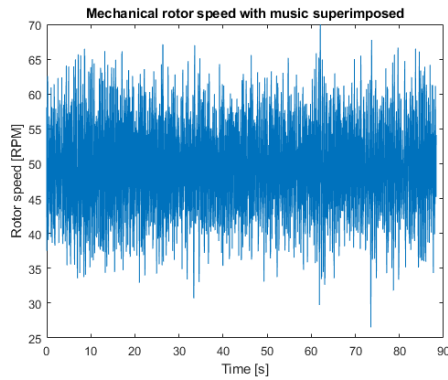


(b) U_D with superimposed music

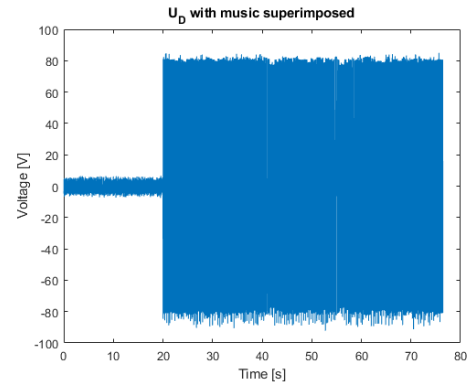
Figure 4.5: Mechanical rotor speed from the machine from the non-rotational case when song is superimposed on U_D

4. Measured sound from the test rigs

For the rotational case, the rotor speed was set to 50 RPM and the dynamometer was not connected since there was no cooling. As can be seen in Figure 4.6 playing a song has less impact on the machine. As can be seen in the figure there is no difference in the speed between 0-20 seconds when no song is playing and 20-70 where the song has started. The rotor speed is equally noisy with and without music. The superimposed voltage from the music is clearly seen in the d-voltage, where the voltage increases when the music starts at 20 s.



(a) Mechanical rotor speed



(b) U_D with superimposed music

Figure 4.6: Mechanical rotor speed from the machine when the rotational case was performed and a song was superimposed on U_D

4.2 Measurements using the stop-switching method

At standstill, the stop-switching method also produced audible differences between the requested frequency and the sound the machine made when performing the test with the alternating frequencies described in Section 3.2.1. But, the frequencies were as mentioned before, perceived as much cleaner for this method compared to the superimposed method. This means that it sounded more like one tone and not a cluster of frequencies. The result became that it was easier to recognize what song that was played both at standstill and in the rotation case. As mentioned in Section 3.3 the stop switching method has a dynamic bandwidth gain to handle the loss of voltage during the non conductive states. The values of the gain shown in Table 3.3 are plotted in Figure 4.7 as dots. As mentioned before the gain function shown as a line in Figure 4.7 were derived with cftool and the result became:

$$f(x) = 3.271 \cdot 10^{-06} \cdot x^{-1.481} + 1.015 \quad (4.1)$$

where the x is the period time of the frequency played. The gain function is then multiplied with the bandwidth of the controller to compensate the loss of voltage.

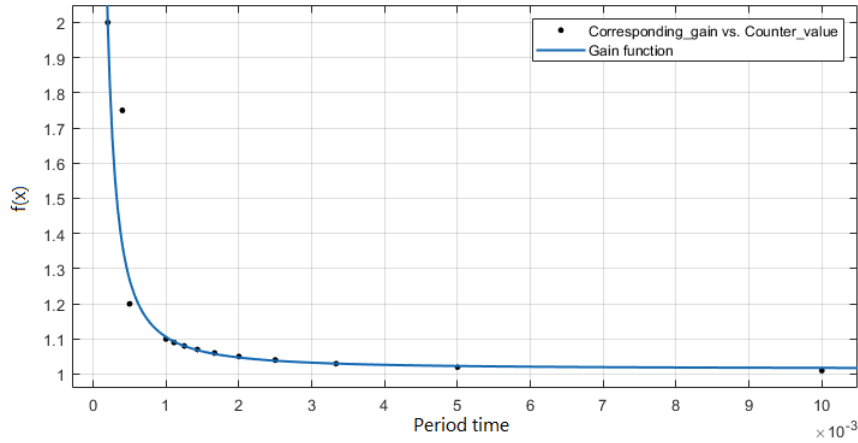


Figure 4.7: Dynamic gain function

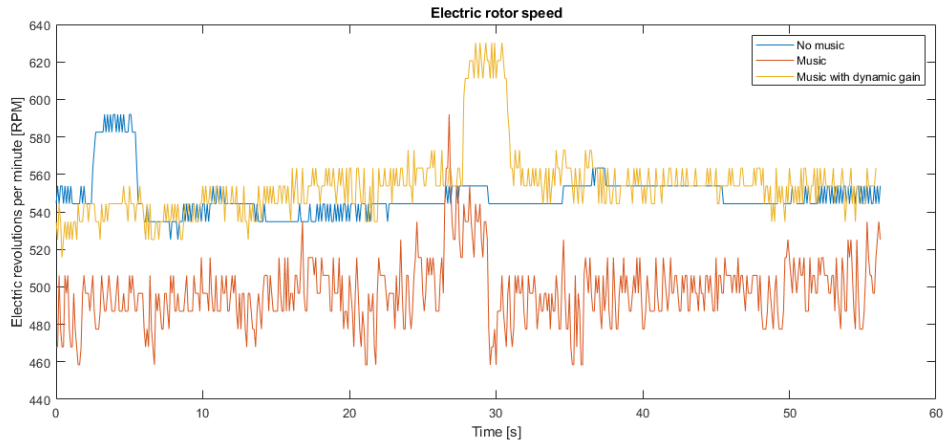
To compare the stop-switching method to the superimpose method and also to evaluate the dynamic gain function two different tests were done. For both tests 3 A were requested in both I_q and I_d and the result can be seen in Figures 4.1 and Figure 4.8. As one can see, the music impacts the rotor speed slightly when there is no dynamic gain. As mentioned the measurement units are designed to measure currents up to 700 A peak, while the most current that was requested during these tests were 3-5 A peak. Since the increase is a percentage increase based on the time the driver is in a passive zero mode, the gain for low signals will be very low. Figure 4.7 shows that the maximum gain is reached at 5000 Hz which doubles the gain. Some of the frequencies that are played are shown in Table 3.1 and they are around 700 Hz which, if converted to the period time and is inserted in (4.1) gives:

$$3.271 \cdot 10^{-06} \cdot 0.0014^{-1.481} + 1.015 = 1.068 \quad (4.2)$$

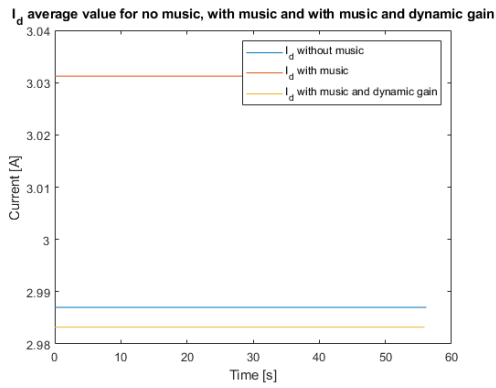
An increase of 6.8% which multiplied with the bandwidth gives:

$$1.068 \cdot 5 = 5.34 \quad (4.3)$$

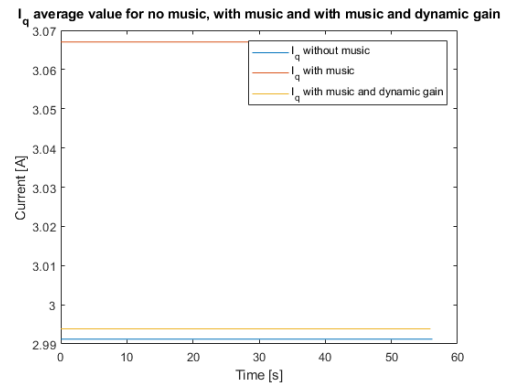
The measurement noise is probably larger than the effect of that increase in gain which negate the effect of the dynamic gain.



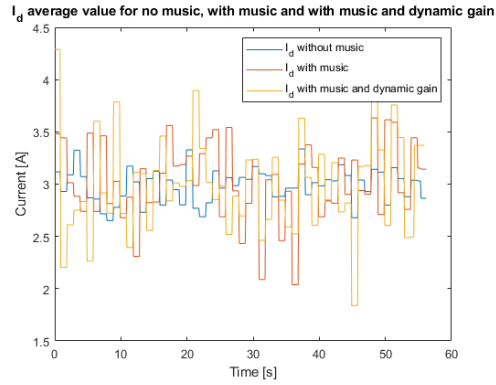
(a) Electrical rotor speed comparison between all three sub-methods



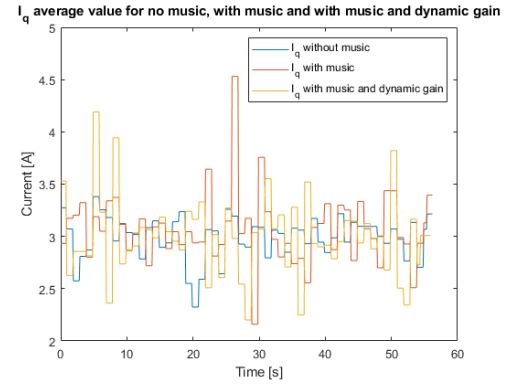
(b) Id average value over entire measurement period



(c) Iq average value over entire measurement period



(d) Id average value over ten samples entire measurement period



(e) Iq average value over ten samples entire measurement period

Figure 4.8: Comparison for the rotating case when stop-switch method was performed.

It is not clear if the average current is affected by the stop-switching method by looking at the current measurements. In figure 4.8 (b) and (c) one can observe that the difference is around 0.07 A at most and as mentioned the current measurement units needs at least 0.4 A to measure the difference. There is a factor 5.7 between these and therefore the range can be considered to be within the measurement uncertainty. This result is therefore considered acceptable for the goal of not affecting the average current which is the goal of the thesis. By looking at Figure 4.8 (a) one can observe that there is a notable speed difference and when music is added the speed is lowered. This can be explained by the removal of voltage pulses which creates the sound. Since there is no compensation for that it is reasonable for the speed to be lower. The speed is about the same when looking at the dynamic gain case which is interesting, since the bandwidth already was very low, the impact of the dynamic gain would be very small. At around 25 seconds the speed increases almost 100 electrical RPM in both the dynamic gain case and music case, which indicates that some frequencies causes the rotor to rotate faster.

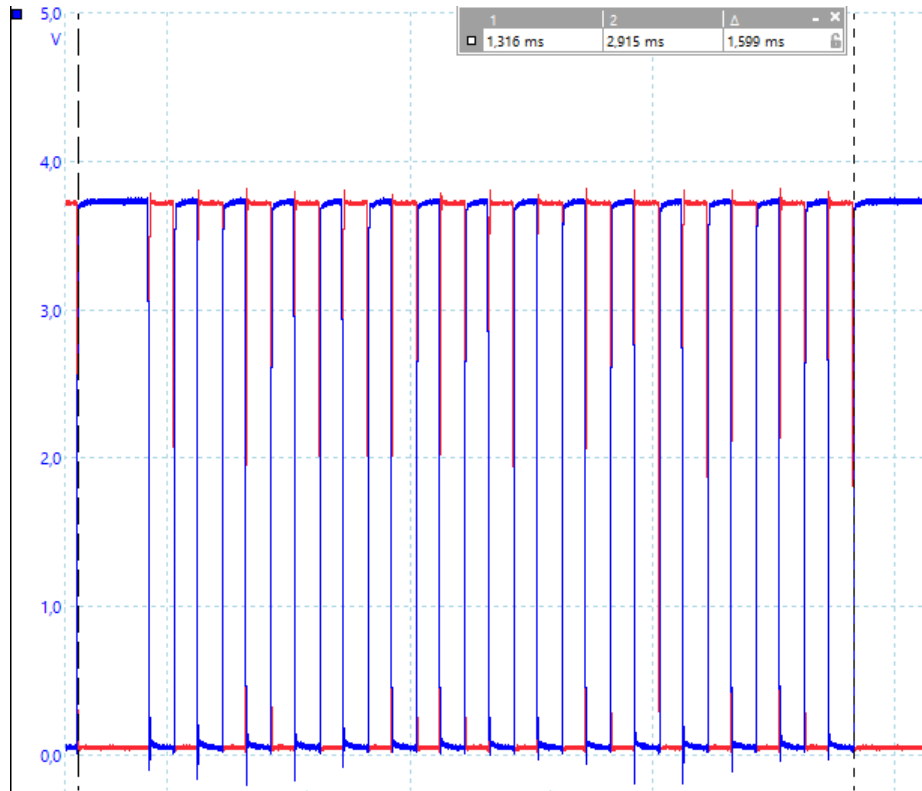


Figure 4.9: PWM-pattern from one phase leg with song playing

The PWM pattern in Figure 4.9 is the gate driver signals to one phase leg of the inverter when a song is playing with the stop-switching method. Blue is channel A and is showing the signal that going to the lower IGBT while red is channel B and is showing the signal to the upper IGBT. The reason why the blue signal is on when it should be off is that it is set that way in the drivers. Which means that they both are off. One can observe that there are pulses missing. Which supports that the software works as intended and that it is the missing pulses that are creating the tones that are generated. The capture is taken when the song has started. The first frequency played is 660 Hz and the expected behaviour is that there should be a pulse missing every 15ms. The figure says 1.599ms, which is 625 Hz. This is the frequency which hits the PMSM, to find out which frequency that results in when listening the acoustic transfer function must be considered.

4.3 Acoustic transfer function of the lab-rig PMSM

The acoustic transfer function of the PMSM was extracted with the method described in Section 3.4 for both the stop-switching method and the superimpose method for the non-rotational case. The acoustic transfer function from the stop-switching method was extracted by playing each frequency for fifteen seconds while recording the sound from the PMSM with MATLAB and a SONY MDR-XB950N1 headset as microphone. The measurement was processed in a MATLAB-script that did an FFT on each vector to obtain the frequency content of the recorded sound. As mentioned before, the algorithm selects the most dominant tone as the one that is presented. The acoustic transfer function obtained by using this method can be seen in Figure 4.10

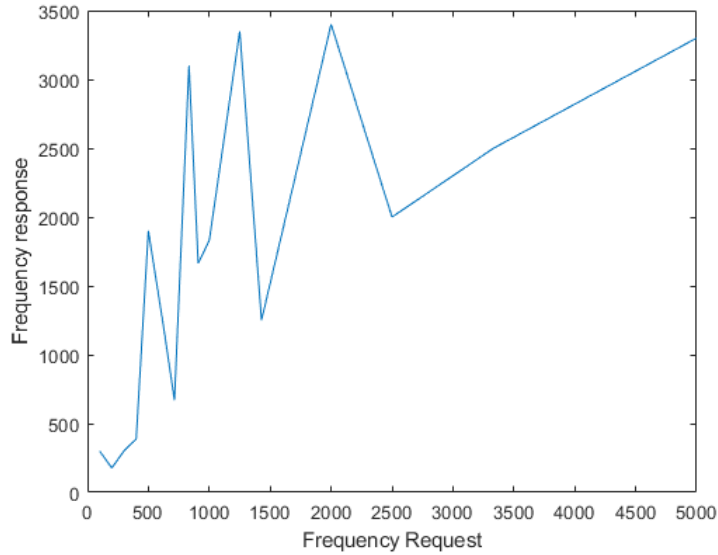
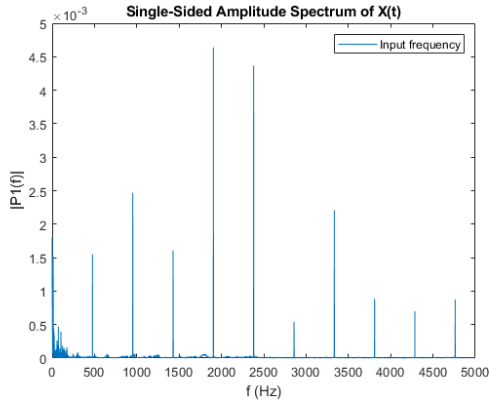
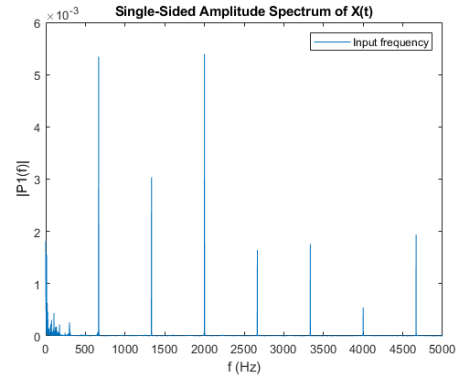


Figure 4.10: The acoustic transfer function of the PMSM when using stop-switching method and the PMSM is stationary

As can be noted, the transfer function is very non-linear and that depends on that there are several other frequencies recorded that in some cases are louder than the frequency played. Two examples of the FFT from the tests can be seen in Figure 4.11, where the entire frequency spectrum that is returned from the PMSM is shown for when a frequency of 500 Hz and 700 Hz were requested. As one can see, in Figure 4.11 (a) there is an energy spike at 500Hz, which is the requested frequency. But since the algorithm looks for the single largest value it finds the value that is around 1800 Hz instead, which also can be seen in Figure 4.10. In Figure 4.11 (b) the response from a request of 700 Hz is shown and the energy spike is at 667 Hz. Note that when 500 Hz is requested there is no spike at 700 Hz and vice versa. Which means that the transfer function is somewhat linear and the PMSM returns almost what is requested but not always as the dominant frequency.



(a) Spectrum of requested 500 Hz



(b) Spectrum of requested 700 Hz

Figure 4.11: Frequency spectrum of the recorded sound form the PMSM when 500 Hz and 700 Hz were requested.

To verify the results from the measurement with the SONY MDR-XB950N1 headset the same measurement was done with the SQuadriga equipment. The SQuadriga measurements were processed in another software that accompanies the SQuadriga, called PAK Software by Mueller-BBM. For the SQuadriga measurement when doing the stop-switching test each frequency vector was played for ten seconds instead of fifteen. The time reduction was done since there was a theory that there would be different harmonics when transition from frequency to frequency. However, that effect was not observed which means that there was no audible noise in-between transitions and therefore the time was cut by one third.

The result for the stop-switching method from the SQuadriga measurements can be seen in Figures 4.12, 4.13, 4.14 and 4.15 where the X-axis shows time and the Y-axis frequency. The color in the graph represents the sound intensity and the intensity scale in dB is shown on the right side of the graph. In Figure 4.12 and 4.14 frequencies between 100-800 Hz were tested according to Table 3.4 and they are played for 10 s each (0-10 s 100 Hz,..... 70-80 s 800 Hz). In Figure 4.13 and 4.15 frequencies between 909-5000 Hz were tested according to Table 3.4 and they are played for 10 s each (80-90 s 909 Hz,..... 160-170 s 5000 Hz). In Figures 4.12 and 4.13 the blue color is for the lowest intensity and should be interpreted as no sound at that frequency and time and the other colors indicate that there is a sound at that frequency at that time. The text boxes in Figures 4.12 and 4.13 marks the frequency with the highest intensity for the respectively 10 s each reference frequency is played.

4. Measured sound from the test rigs

By comparing Figure 4.11 (a) with Figure 4.12 between 40 s and 50 s when 500 Hz is played, it can be noticed that two spikes in Figure 4.11 (a) corresponds to the two lines in Figure 4.12. One slightly below 2000Hz and the other one slightly below 2500Hz. However, the most dominant tone is different in both of the cases. The test were repeated to confirm the most dominate and the result can be seen in Figure 4.14 and 4.15. As can be noted the dominate tone is a different one which proves that the dominate tone is quite sporadic. The green noise, which is a result of the clicking noise the that comes from the electric machine when it's not allowed to spin is also quite sporadic but appears around 800 Hz and continues up to 2000 Hz.

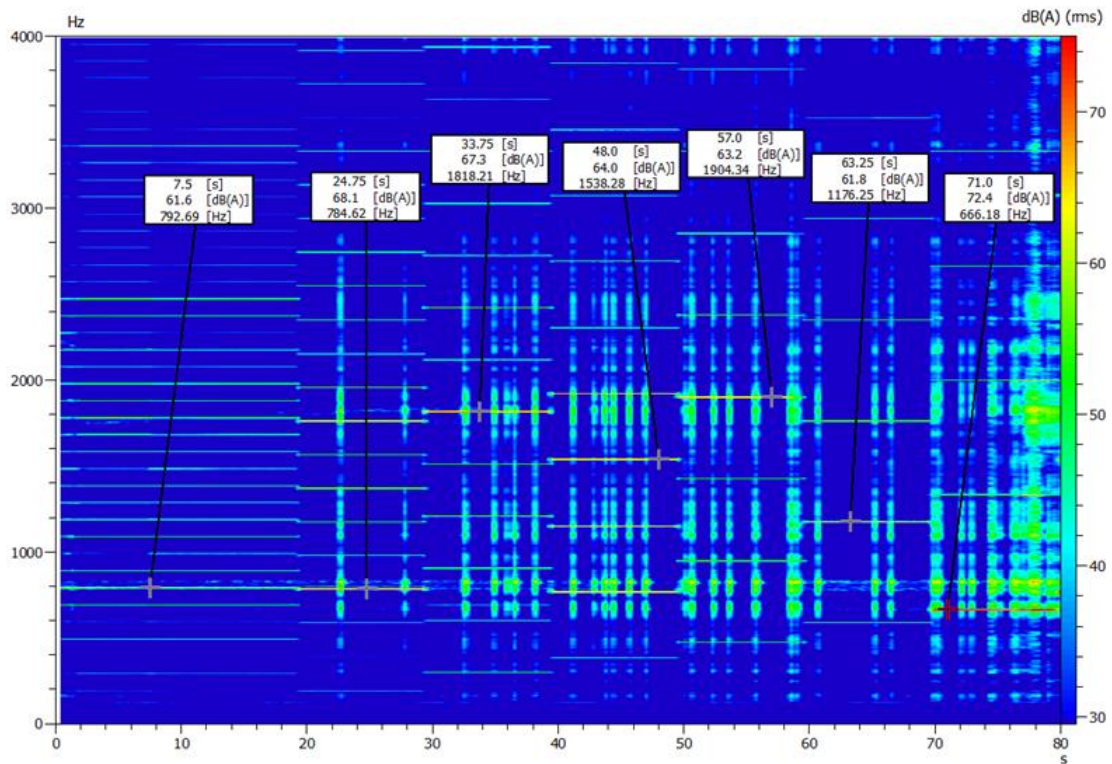


Figure 4.12: Result from the SQuadriga measurements analyzed with the PAK software for the stop-switching method playing single frequencies and the PMSM is non-rotating. The requested frequencies are 100, 200, 303, 400, 500, 625, 714 and 800 Hz for 10 s each (0-10 s 100 Hz,..... 70-80 s 800 Hz)

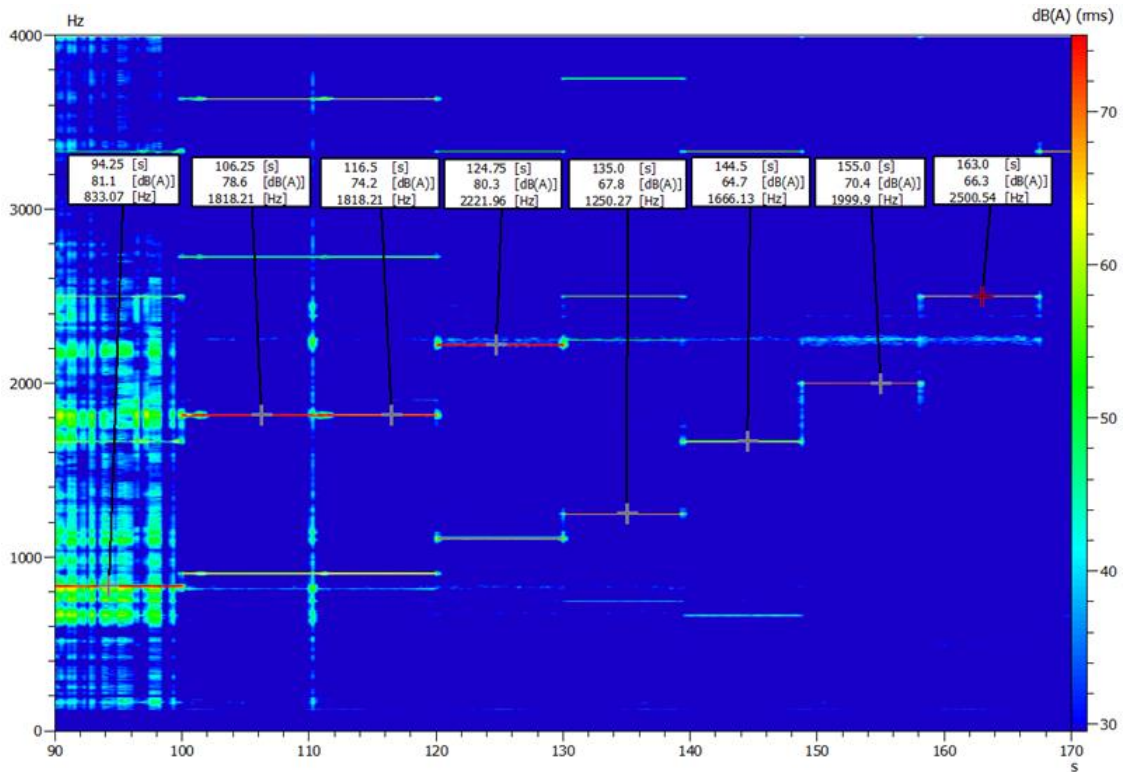


Figure 4.13: Result from the SQuadriga measurements analyzed with the PAK software for the stop-switching method playing single frequencies and the PMSM is non-rotating. The requested frequencies are 909, 1000, 1250, 1428, 2000, 2500, 3333 5000 Hz for 10 s each (80-90 s 909 Hz,..... 160-170 s 5000 Hz).

Something that can be seen in all cases is that the requested values are seen like a staircase in 0-80 seconds, it's clear in Figure 4.14, in Figure 4.12 it's harder to spot but it's present even there. The frequencies in the stair is close to the requested value, it is hard to say exactly what frequency it is since they are not pin pointed, they are likely below the requested value since that is something that can be seen in Figure 4.11. The staircase then continues in Figure 4.13 but these frequencies are not the requested values. The value between 160-170s 2500 Hz is half of what is requested, 5000 Hz. The frequencies are not the requested ones but they still increase when the request increase. This is somewhat reasonable, since the transfer function of the electrical machine will affect the output frequency. The result also shows a linearity in both cases in higher frequencies. It is hard to spot in Figure 4.15 but the three last most dominate tones are the same in both cases which corresponds to what can be seen in 4.10 That when frequencies over 3000 Hz is requested it's a linear response. When comparing Figure 4.11 (b) with Figure 4.12 between 70 s and 80 s (when 700 Hz is played) it is more difficult to see the similarities due to that the plotting of the SQuadriga measurements and the MATLAB measurement are different and also the clicking noises are sporadic. This distortion could be mechanical properties of the ferromagnetic material the stator is made of, it is excited during these reference frequencies and not during the other.

4. Measured sound from the test rigs

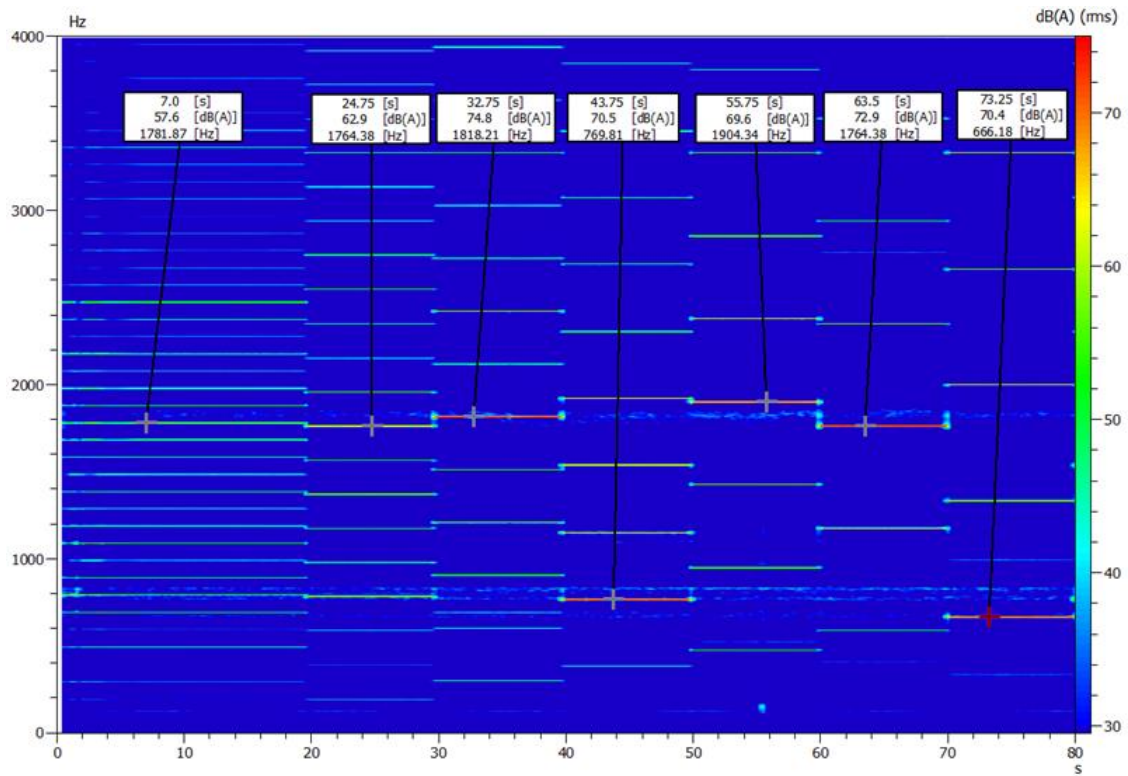


Figure 4.14: Result from the SQuadriga measurements analyzed with the PAK software for the stop-switching method playing single frequencies and the PMSM is non-rotating. The requested frequencies are 100, 200, 303, 400, 500, 625, 714 and 800 Hz for 10 s each (0-10 s 100 Hz,..... 70-80 s 800 Hz)

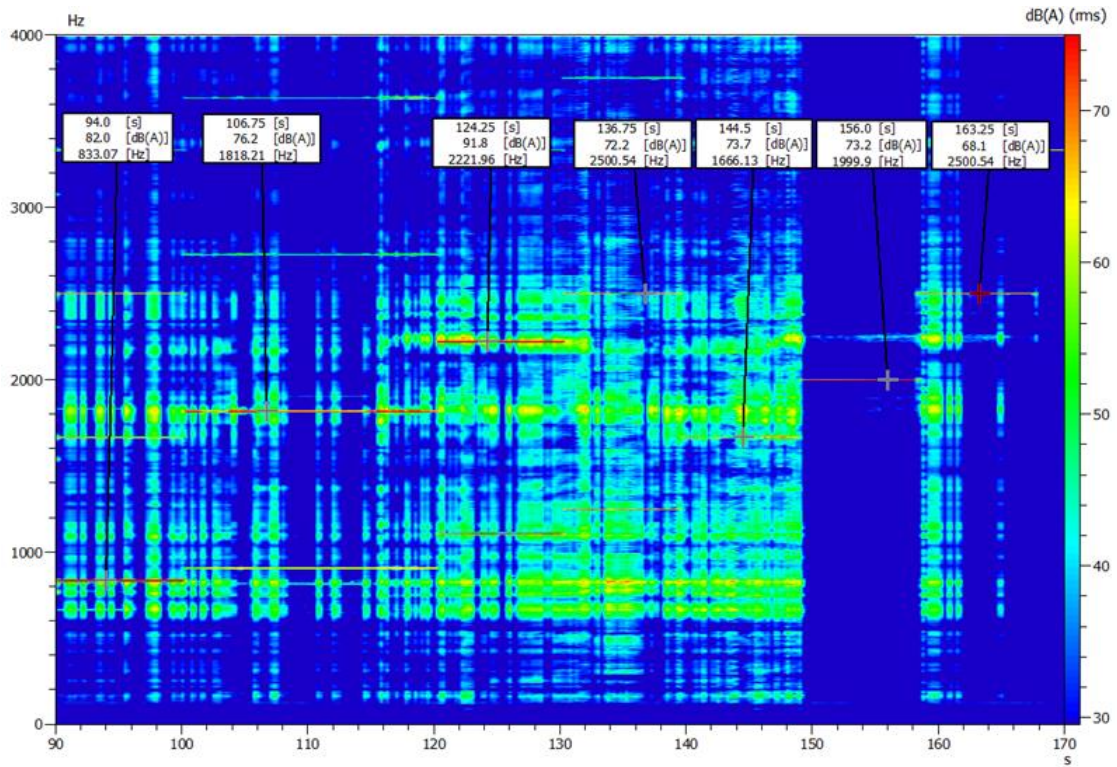


Figure 4.15: Result from the SQuadriga measurements analyzed with the PAK software for the stop-switching method playing single frequencies and the PMSM is non-rotating. The requested frequencies are 909, 1000, 1250, 1428, 2000, 2500, 3333 5000 Hz for 10 s each (80-90 s 909 Hz,..... 160-170 s 5000 Hz).

The transfer function for the superimpose method measured with the SQuadriga equipment can be seen in Figure 4.16, where the X-axis shows time and the Y-axis frequency. The color in the graph represents the sound intensity and the intensity scale in dB is shown on the right side of the graph. In this figure all frequencies in Table 3.4 are played for 5 s each (0-5 s 100 Hz, 5-10 s 200 Hz,....., 70-75 s 3.3 kHz), in addition 5 kHz is played between 75 s and 80 s and the reference frequencies are marked with red ellipses. As can be noted in the figure it's a lot of harmonics and there is not a clear input to output transfer function. There is a similarity between the methods that the correct tones are present in the figures as stairs. In Figure 4.16 The stairs starts in the bottom left corner of the figure after the ten second mark. The frequency between ten and fifteen seconds should be 300 Hz according to table 3.4, the stair is about 200 Hz and then increases 100 Hz every five seconds. However, it can be a coincidence, almost any pattern that is wanted can be found when there is so many measurement points, for example can twice the frequency also be found for almost all frequencies played which is interesting since, as mentioned in the Section 2.3.4 the frequency should be twice the frequency if a ferromagnetic material is subject of an AC-field. In Figure 4.11 (a) and (b) twice the requested frequency can also be seen. The same effect can be seen in the superimpose case but the same conclusion can be drawn that it can be noise that happens to fit the theory.

The marks in red are the requested values and as can be seen in two places 1250 Hz and 2500 Hz there is no sound. When listening at the recording there is no sound in that section either, indicating that it is hard for this inverter to produce those frequencies on this PMSM. That can depend on material in the stator, size of the machine or not enough current to produce those frequencies.

Something that both methods resulted in but cannot be seen in the graphs was that the frequency response was changed a bit depending on the rotor position. The music played when the rotor rotated therefore became more consistent in the tones they produced. This effect was only noticed on the lab-rig and it is possible that it is machine dependent.

That the transfer functions looks different for the two different methods can depend on that a sinus signal is rather soft while a stop switch causes an abrupt turn-off like a square wave. A square wave have a lot of harmonics and are therefore unpredictable. There was no way of measuring the phase current with high resolution at the lab rig, otherwise the phase current harmonics could be related to the harmonics produced by the motor, to see what the mapping actually are and if it is consistent i.e. If a measured current signals shows a harmonic of 1000 Hz, if that would be a resulting 1000 Hz in sound or similar.

It can also be a matter of size and current, for the superimpose method on the lab rig, the current request were rather low since if higher current were requested the rotor began to rotate, it is possible that higher current request with a locked rotor scenario would produce cleaner tones with less noise. Now, a acoustic transfer function was never done on the production rig but when doing the superimpose on

the production rig 70 V was requested and a resulting I_d of 200 A was supplied to the machine. 200 A causes a relatively strong magnetic field and from there the cleaner perceived tones could be heard.

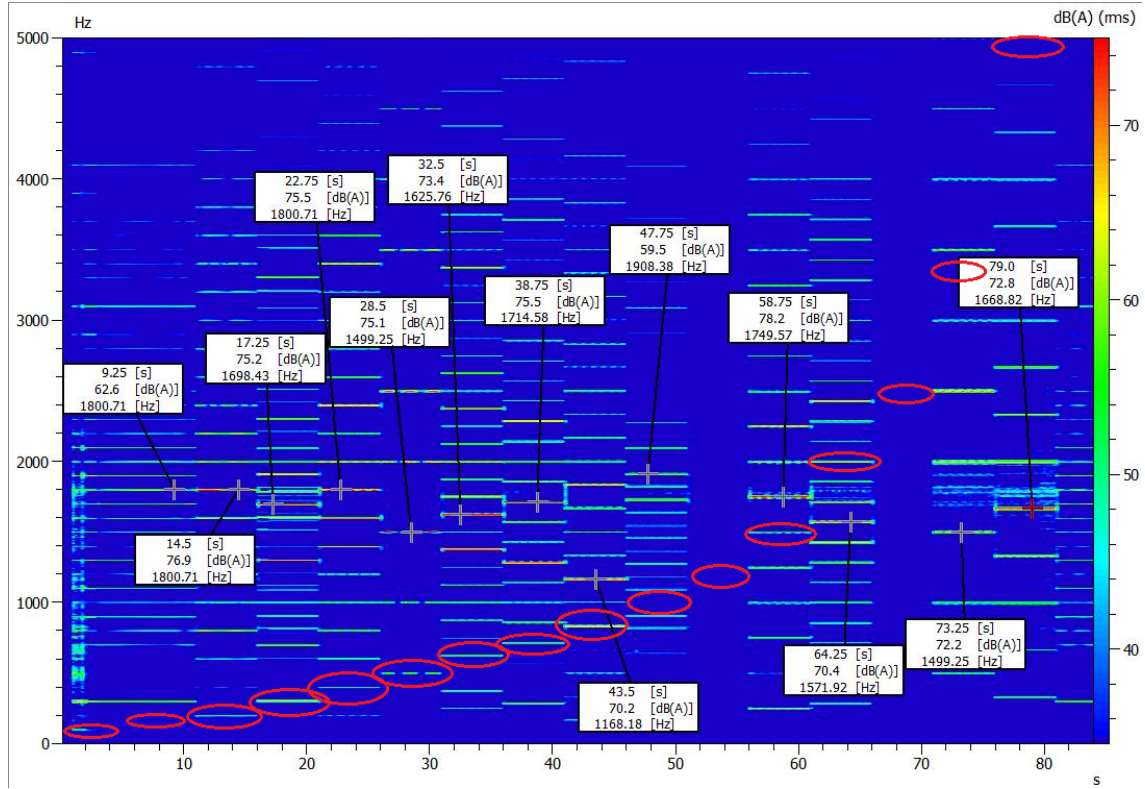


Figure 4.16: Result from the SQuadriga measurements analyzed with the PAK software for the superimpose method playing single frequencies and the PMSM is non-rotating. The requested frequencies are shown in Table 3.4 and they are played for 5 s each (0-5 s 100 Hz, 5-10 s 200 Hz,....., 70-75 s 3.3 kHz) and in addition 5 kHz is played between 75 s and 80 s.

4. Measured sound from the test rigs

5

Conclusion

The conclusions which can be drawn from this thesis is that it is possible to take advantage of the magnetostriction effect to produce audible sounds with an electric machine. The audible sounds can be produced without affecting the average torque within the boundaries of measurement error. Of the two evaluated methods the stop-switching method sounds better then the superimpose method on the lab rig. For the superimpose method, it sounds better on the production rig compare to the lab rig.

The acoustic transfer function of the lab-rig PMSM depends both on the method used and rotor position. The frequency response is creates an audible difference when listening and a visible distance of change in frequency spectrum when recorded and analyzed

Even if the rise time of the controller is lower than the lowest frequency that is superimposed, the rotor still moves but produces low torque.

5.1 Future work

Extend the range of frequencies which can be played with the stop-switching method by switching from time periods to an on-off-method where a modifiable amount of pulses can be on and off instead of a set number.

Adjust dynamic gain function with respect to voltage limiter to prohibit reaching the limit.

Find out maximum voltage that is needed for superimpose, since saturation hits quite fast. Make it dynamic by change it with the relation to the voltage limiter

Find out the acoustic transfer function for the rotating case.

Find out the wear on the electric machine and inverter when used as speaker.

Test the methods in a real car.

Bibliography

- [1] J.Ekblom, "Denmark calls for EU strategy to phase out diesel and petrol cars from 2030", Reuters,OCT 4, 2019,[Online]. Available: <https://www.reuters.com/article/us-eu-autos-denmark-idUSKBN1WJ1YW> [Accessed June. 5, 2021].
- [2] J. Porter, "Fake noise will be added to new electric cars starting today in the EU ," The Verge, Jul 1, 2019. [Online]. Available: <https://www.theverge.com/2019/7/1/20676854/electric-cars-artificial-safety-noise-low-speeds-european-union-rules-2019-2021> [Accessed March. 1, 2021].
- [3] <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811526>
- [4] Wikipedia contributors. (2021, May 5). Electric vehicle warning sounds. In Wikipedia, The Free Encyclopedia. Retrieved 10:57, June 10, 2021, from https://en.wikipedia.org/w/index.php?title=Electric_vehicle_warning_sounds&oldid=10216165
- [5] B.D.Cullity, "Fundamentals of Magnetostriction." JOM 23, 35–41 (1971). <https://doi.org/10.1007/BF03355677>
- [6] Raymond A. Serway and John W. Jewett, "The Laws of Motion", Physics for scientists and Engineers with modern physics, 9th ed Boston, MA, USA: Cengage Learning, 2015.
- [7] <https://www.britannica.com/science/electromagnetism/Magnetic-fields-and-forces>
- [8] Britannica, T. Editors of Encyclopaedia (2019, February 21). Magnetic dipole. Encyclopedia Britannica. <https://www.britannica.com/science/magnetic-dipole>
- [9] H. Chang, "Reliable IGBTs for hybrid/electrical vehicles (H)EV," 2014 IEEE International Integrated Reliability Workshop Final Report (IIRW), 2014, pp. 165b-165b, doi: 10.1109/IIRW.2014.7049541.
- [10] Mohan, Ned Undeland, Tore M. Robbins, William P.. (2003). Power Electronics - Converters, Applications, and Design (3rd Edition). (pp. 227,626). John Wiley Sons. Retrieved from <https://app.knovel.com/hotlink/toc/id:kpPECADE0Q/power-electronics-converters/power-electronics-converters>
- [11] Ali, W.H., Sadiku, M.N.O., Abood, S. (2019). Fundamentals of Electric Machines: A Primer with MATLAB: A Primer with MATLAB (1st ed.). CRC Press. <https://doi.org/10.1201/9780429290619>
- [12] <https://www.sciencedirect.com/topics/chemistry/magnetostriction>

- [13] S. Lundberg, Lecture 3 and 4, PMSM modelling, transformation theory. Lecture notes, Course: ENM076 Electric drive systems, Chalmers University of Technology, 2021.
- [14] Yujing Liu, PMSM, Lecture slides, Course: Electrical Machines -Design and Analysis, Chalmers University of Technology, 2020.
- [15] W. C. Duesterhoeft, M. W. Schulz and E. Clarke, "Determination of Instantaneous Currents and Voltages by Means of Alpha, Beta, and Zero Components," in Transactions of the American Institute of Electrical Engineers, vol. 70, no. 2, pp. 1248-1255, July 1951, doi: 10.1109/T-AIEE.1951.5060554.
- [16] R.H. Park Two Reaction Theory of Synchronous Machines AIEE Transactions 48:716–730 (1929).
- [17] Purves D, Augustine GJ, Fitzpatrick D, et al., editors. Neuroscience. 2nd edition. Sunderland (MA): Sinauer Associates; 2001. The Audible Spectrum. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK10924/>
- [18] J. Arellano-Padilla, M. Sumner, C. Gerada, and G. Asher, "Prediction of Inductance Characteristics of PMSMs in Saliency-based Sensorless Control", presented at the Power Electronics and Motion Control Conference, Novi Sad, 2012, LS3a–1.1–1 –LS3a–1.1–5.
- [19] Bengtsson, Lars, Lars Bengtsson, and Bill Karlström. Transformer Och Filter. 2., [Utök.] uppl. Studentlitteratur, 2016.
- [20] Hansen, Eric W., and Eric W Hansen. Fourier Transforms : Principles and Applications, John Wiley Sons, Incorporated, 2014. ProQuest Ebook Central, <https://ebookcentral.proquest.com/lib/chalmers/detail.action?docID=1810513>.
- [21] Wikipedia contributors, "JTAG," Wikipedia, The Free Encyclopedia, <https://en.wikipedia.org/w/index.php?title=JTAG&oldid=1025425115> (accessed June 11, 2021).
- [22] <https://se.mathworks.com/help/matlab/ref/audioread.html>
- [23] https://se.mathworks.com/help/matlab/import_export/record-and-play-audio.html
- [24] <https://se.mathworks.com/help/curvefit/curvefitting-app.html>

A

Appendix 1

A.1 Code

```
1
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %
4 % Program that converts mp3 songs to a Frequency vector %
5 %
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 %
8 % V.4 Improvement of V.3 where frequencies will be mapped %
9 % to avaiable frequencies. %
10 %
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12
13 % Clear all to avoid errors
14 clc
15 clear
16
17 %Load song
18
19 [y,Fs] = audioread("Train.mp3");
20 y = y(:,1); % Load the song into a variable
21 y = y(1732:end);
22 T = 1/Fs; % Sampling period
23 L = length(y); % Length of signal
24
25 % 1 sample on controller 100 us
26 sampleTime = 1/10000;
27
28 % LOOP 5.0
29 % This loop will try to find the frequency of a small part
    of the song and
30 % write that frequency to a new vector and create a time
    compensated vector.
31
32 % Set up variables
```

```

33 mod = 1; % Mod or g could be repaced by each other but are
    kept to increase readability
34 g = 1;
35 samples = 13230/2;
36 freq = 0;
37 prevFreq = 0;
38 signalRange = 100; % 100 will result in 100Hz as lowest
    signal
39 modMax = floor(length(y)/samples);
40 captureTime = T*samples;
41 tempFreqVector = zeros(1,samples); % Pre-allocate vectors to
42 frequencyVector = zeros(1,modMax); % make program run faster
43 timeVector = zeros(1,modMax);
44 AvailableFrequencies = zeros(1,signalRange);
45 clockStep = 1/10000;
46
47
48 for i = 1:signalRange
49     AvailableFrequencies(i) = clockStep;
50     clockStep = clockStep + 1/10000;
51 end
52
53 AvailableFrequencies = 1./ AvailableFrequencies; % This
    vector includes all generatable frequencies from 10kHz to
    100Hz
54
55 for i = samples:samples:L % Loop from samples to end of
    music vector
56     %Determain frequency
57     if((i/samples) == mod) % Modelo function
58
59         if(mod == 1)
60             for x = g:g*samples % Pick samples from the
                correct place from the song
61                 tempFreqVector(x) = y(x); % Write those
                    samples to a temporary vector
62             end
63         elseif(mod == modMax)
64             for x = L*samples:1:(L-samples*modMax) % Pick
                samples from the correct place from the song
65                 tempFreqVector(x) = y(x);
66             end
67         else
68             for x = (g*samples):((g+1)*samples) % Pick
                samples from the correct place from the song
69                 tempFreqVector(x) = y(x);

```



```

70         end
71     end
72
73     xdft = fft(tempFreqVector); % Do a fft of that
        vector
74     [maxval,idx] = max(abs(xdft)); % Find the maximum
        value of the vector and the position of that
        value
75     freq = (Fs*(idx-1))/(length(x)* mod * samples); % I
        have no idea but it works
76     if (g == 1)
77         if(freq == 0)
78             frequencyVector(g) = 0;
79         else
80             frequencyVector(g) = (1/freq);
81         end % If it's the first time we're writing to
            the vector
82     elseif(freq==0 || freq<200) % Check if value is
        equal to 0, write zero to avoid dividing by 0 and
        end the universe
83         frequencyVector(g) = 0;
84     elseif(freq>10000) % Check if value is equal to
        0, write zero to avoid dividing by 0 and end the
        universe
85         frequencyVector(g) = 1/10000;
86     else
87         %Re-map to closest available freq
88         [minDistance, indexOfMin] = min(abs(
            AvailableFrequencies-freq));
89         frequencyVector(g) = 1/AvailableFrequencies(
            indexOfMin);
90     end
91
92
93     % Create the time vector
94     timeVector(g) = (samples/Fs)*g;
95     % Reset variables
96     xdft = 0;
97     tempFreqVector = 0;
98     % Increase variables for correct placement in vector
99     mod = mod + 1;
100     g = g + 1;
101
102 end
103 end
104

```

```

105 % Adjust frequencies
106 frequencyVector = rmmissing(frequencyVector); %For exporting
    the vector
107
108
109 frequencyVector = frequencyVector';

1 %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

2 %
    %
3 % Program that converts mp3 songs to a Frequency vector
    %
4 %
    %
5 %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

6 %
    %
7 % V.6 Sampels dynamically
    %
8 %
    %
9 %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

10
11 % Clear all to avoid errors
12 clc
13 clear
14
15 %Load song
16
17 [y,Fs] = audioread("DoomHangar.mp3");
18 y = y(:,1); % Load the song into a variable
19 %y = y(1732:end);
20 T = 1/Fs; % Sampling period
21 L = length(y); % Length of signal
22
23 % 1 sample on controller 100 us
24 sampleTime = 1/10000;
25
26 % LOOP 5.0
27 % This loop will try to find the frequency of a small part
    of the song and

```

```
28 % write that frequency to a new vector and create a time
    compensated vector.
29
30 % Set up variables
31 mod = 1;

    % Mod or g could be repaced by each other but are kept to
    increase readability
32 g = 1;
33 samples = 13230/2;
34 freq = 0;
35 prevFreq = 0;
36 signalRange = 100;

    % 100
    will result in 100Hz as lowest signal
37 modMax = floor(length(y)/samples);
38 captureTime = T*samples;
39 tempFreqVector = zeros(1,samples);

    % Pre-allocate vectors
    to
40 frequencyVector = zeros(1,modMax);

    % make program run
    faster
41 timeVector = zeros(1,modMax);
42 AvailableFrequencies = zeros(1,signalRange);
43 clockStep = 1/10000;
44
45
46 % plot(y)
47 % xlabel('Samples')
48 % ylabel('Amplitude')
49
50 %%
51
52 for i = 1:signalRange
53     AvailableFrequencies(i) = clockStep;
54     clockStep = clockStep + 1/10000;
55 end
56 % Start sample, assume values will increase, when values
    decrease assume
57 % they will go to zero abs(0.05) is first limit
58
59
60 dynamicZero = 0.035; % THIS VALUE MUST BE FOUND FOR EACH
    SONG BY LOOPING THE SONG AND FIGURE OUT THE VALUE
61 numberOfZeros = 150;
```

```

62 zeroCounter      = 0;
63 toneSet          = 0;
64 valueCounter     = 0;
65 saveZero         = 0;
66 avarageCounter   = 0;
67 avarageValue     = 0;
68 dynamicSampling = zeros(1,2000); %Change singal range to
    correct value later
69 j = 1;
70 avarageThreshold = 100;
71 avarageVector = zeros(1,floor(L/avarageThreshold));
72 %This loop will go through the song and figure out when a
    tone is played
73 %and when thre is a pause in between tones. It will do so by
    first assume 0
74 %until a tone starts, the tone starts by that the value of
    the song is
75 %higher than the treshhold, then the loop wait for a lot of
    zeros, since
76 %the values
77 figure(1)
78 plot(y)
79
80 for i=1:L
81     % Read value
82     value = abs(y(i));
83
84     avarageCounter =avarageCounter+ 1;
85     avarageValue = avarageValue+ value;
86
87     if(avarageCounter == avarageThreshold)
88         avarageVector(j) = avarageValue/avarageThreshold;
89         avarageCounter = 0;
90         avarageValue = 0;
91         j = j+1;
92     end
93
94     if(value<dynamicZero)
95         value = 0;
96         zeroCounter = zeroCounter + 1;
97         valueCounter = 0;
98
99     % Assume song has started
100    elseif(g == 1)
101        dynamicSampling(g) = i; % Save pos when song has
            started

```

```

102         zeroCounter = 0;           % reset zero counter
103         figure(2)
104         plot(y(1:dynamicSampling(1)))
105         g = g + 1;
106         % Tone has been played but a pause has also been
           intitiated.
107         elseif(zeroCounter > numberOfZeros && toneSet == 0)
108             dynamicSampling(g) = i-sum(dynamicSampling(1:(g)))-
               zeroCounter; % saves samples that are the tone
109             figure(3)
110             plot(y(sum(dynamicSampling(1:g-1)):sum(
               dynamicSampling(1:g))))
111             title('Tone period')
112             toneSet = 1;
113             g = g+1;
114
115         % HERE G>1 and a tone has just started
116         else
117             if(saveZero == 0 && toneSet == 1)
118                 saveZero = zeroCounter;
119             end
120             valueCounter = valueCounter +1;
121             if((toneSet == 1 && valueCounter>15))
122                 dynamicSampling(g) = saveZero;
123                 figure(4)
124                 plot(y(sum(dynamicSampling(1:g-1)):sum(
               dynamicSampling(1:g))))
125                 title('Zero period')
126                 zeroCounter = 0;
127                 toneSet = 0;
128                 saveZero = 0;
129                 g = g+1;
130
131
132             else
133                 zeroCounter = 0;
134             end
135         end
136
137     end
138     figure(5)
139     plot(avarageVector)
140
141
142
143     %%

```

```

144 resetVar = 0;
145 g = 1;
146 for i = 1:L
147     %disp(i-resetVar)
148     % Loop from samples to end of music vector
149     %Determain frequency
150     if (((i-resetVar)/dynamicSampling(g)) == mod)
                                                    % Modelo
        function
151
152         if(mod == 1)
153             for x = g:g*dynamicSampling(g)
                                                    % Pick
                samples from the correct place from the song
154                 tempFreqVector(x) = y(x);
                                                    % Write
                those samples to a temporary vector
155             end
156         elseif(mod == modMax)
157             for x = L*dynamicSampling(g):1:(L-
                dynamicSampling(g)*modMax) % Pick samples
                from the correct place from the song
158                 tempFreqVector(x) = y(x);
159             end
160         else
161             for x = (g*dynamicSampling(g)):((g+1)*
                dynamicSampling(g)) % Pick samples from
                the correct place from the song
162                 tempFreqVector(x) = y(x);
163             end
164         end
165
166         xdft = fft(tempFreqVector);
                                                    % Do a
                fft of that vector
167         [maxval,idx] = max(abs(xdft));
                                                    % Find the
                maximum value of the vector and the position of
                that value
168         freq = (Fs*(idx-1))/(length(x)* mod *
                dynamicSampling(g)); % I have no idea
                but it works
169         if (g == 1)
170             if(freq < 200)
171                 frequencyVector(g) = 0;
172             else

```

```
173         frequencyVector(g)= freq;
174     end

        % If it's the first time we're writing to the
        vector
175     elseif(freq==0 || freq<200)

        % Check
        if value is equal to 0, write zero to avoid
        dividing by 0
176         frequencyVector(g) = 0; % Limit the lowest
        frequency to 100 Hz
177     elseif(freq>10000) % Check if value is
        equal to 0, write zero to avoid dividing by 0
178         frequencyVector(g) = 10000;
179     else
180         frequencyVector(g) = freq;
181     end
182
183     % Create the time vector
184     timeVector(g) = (dynamicSampling(g)/Fs)*g;
185     % Reset variables
186     xdft = 0;
187     tempFreqVector = 0;
188     % Increase variables for correct placement in vector
189     mod = mod + 1;
190     g = g + 1;
191     resetVar = i;
192
193     end
194 end
195 % Adjust frequencies
196
197 mappingMatrix = zeros(16,2);
198 frequencyRequest =
    [100,200,303.030303030303,400,500,625,714.285714285714,833.333333333333
      1000,1250,1428.57142857143, 2000
      2500,3333.33333333333,5000];
199 frequencyResponse =
    [300,177,305,387,1900,1200,670,3100,1664,1830,3350,1250,3400,2000,2500

200 mappingMatrix(:,1) = frequencyRequest;
201 mappingMatrix(:,2) = frequencyResponse;
202
203
204 frequencyVector = rmmissing(frequencyVector); %For exporting
    the vector
```

205

206

207 `frequencyVector = frequencyVector ';`

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY