Control Algorithm Design for a High Power Electronic Speed Controller

Maarten van Schagen (5360161), Yana Brodskaya (5077400)

June 23, 2023

Abstract

This thesis focuses on finding a control algorithm that, based on a list of requirements, is most suitable for driving the 25 A permanent magnet synchronous motor and propeller that are provided by Fusion Engineering. Various control algorithms are investigated, and basic implementations are simulated. Based on the investigation, it was decided to further optimize the implementation for trapezoidal control and field oriented control. These two algorithms were benchmarked, in order to determine which is more in line with the requirements. This thesis proposes an implementation for a trapezoidal controller that meets most of the requirements.

Preface

This project is part of our bachelor thesis at the Electrical Engineering faculty of the Delft University of Technology. And it would not have been possible without the help of the following people:

- Jianning Dong PhD
  Jianning Dong was our supervisor for this project. Whenever we got stuck, he was always the person to help us, but somehow he was able to do this while letting us stay in full control of the project. He was both the most helpful and nicest supervisor we could have asked for.
- Emeric de Bodin de Galembert, Quinten Luyten, Maxence Simonart, and Ruben Vos
  Emeric de Bodin de Galembert, Quinten Luyten, Maxence Simonart, and Ruben Vos were our groupmates for the overall project. They designed the other modules required to get the project working, and helped us by sharing their findings and their learning experiences.
- Dr.ing. Bart Roodenburg
  Bart Roodenburg helped us with everything that required a setup in the lab. His feedback was always super helpful, and his help with setting up measuring equipment was invaluable.
- Sanjesh Hoskopple, Marwan Hussein, and Tim Koning at Fusion Engineering
  Sanjesh Hoskopple, Marwan Hussein, and Tim Koning proposed this project to us and helped us set the requirements for it. They also gave us tips and feedback where needed and provided us with our budget.
- Dr. Ioan E. Lager and support staff
  Ioan E. Lager is in charge of the Bachelor Thesis course at the TU Delft. He has also helped us with finding a project and supervisor.

We would like to thank all of these people for helping us with our project, we could not have done it without each of you. We have definitely learned a lot from this project; from general skills, like how to design systems, to specific skills like how to tune a trapezoidal controller. And even though the project may have been stressful at times, it was very much worth every last second of it. We are super grateful to have had the chance to get to do this project.

Contents

Chapter 1

Introduction

1.1 Problem definition

The problem that the this project tackles is that there are no electronic speed controllers available for the setup used that have a large enough current rating to allow for a 100% margin, which Fusion Engineering has specified is required for their application. This report solves the problem of driving the inverter, which will be designed by another group, to effectively control the "T motor mn501-s 240kv" motor [1] with the P18×6.1 propeller [2].

1.2 Thesis synopsis

The main design problems this report will tackle are:

1. Which control algorithm fits best for the setup used?
2. How must the tuning parameters be set for this control algorithm?

1.3 State-of-the-art analysis

1.3.1 Global Idea behind motor control of a PMSM

The "T motor mn501-s 240kv" motor is a Permanent Magnet Synchronous Motor (PMSM) with an exterior rotor [1]. Figure 1.1 shows a schematic of the mechanical model of the PMSM. As can be seen, the rotor of a PMSM consists of permanent magnets, which can be divided into pole pairs. One pole pair is formed by a north pole and a south pole. In case of the "T motor mn501-s 240kv" motor, the amount of pole pairs is equal to 14. The stator of the PMSM is formed by motor windings, also shown in Figure 1.1. Please note that Figure 1.1 shows a reduced number of motor windings and pole pairs. The mechanical model of the motor can be converted to a three-phase electrical model, as depicted in Figure 1.2. For the electrical model, the following holds:

$$\omega_{el} = \omega_{mech} \cdot P \tag{1.1}$$

Where P is the number of pole pairs, ω_{el} is the electrical frequency in $\mathrm{rad\,s^{-1}}$ and ω_{mech} is the mechanical frequency in $\mathrm{rad\,s^{-1}}$.

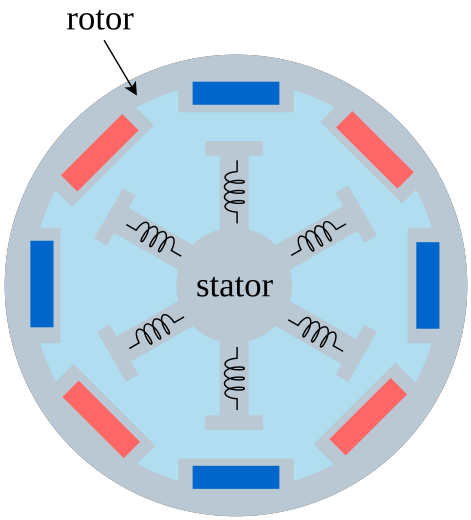

Figure 1.1: A schematic representation of the cross-section of a PMSM with an exterior rotor. A blue block in the rotor represents a south pole, and a red block represents a north pole.

Due to the permanent magnets on the rotor, a rotating magnetic field is present inside a spinning motor. By exciting the motor phases, another magnetic field can be created, which can exert a torque on the rotor and thus make the rotor spin. An example is shown in Figure 1.2: since phase A is connected to the voltage supply, phase B is grounded and phase C is floating, current is flowing from the A phase into the B phase. This results into a magnetic field with a north pole at the A terminal and a south pole at the B terminal. By varying the amount of current flowing through the terminals, the direction of the magnetic field can be changed.

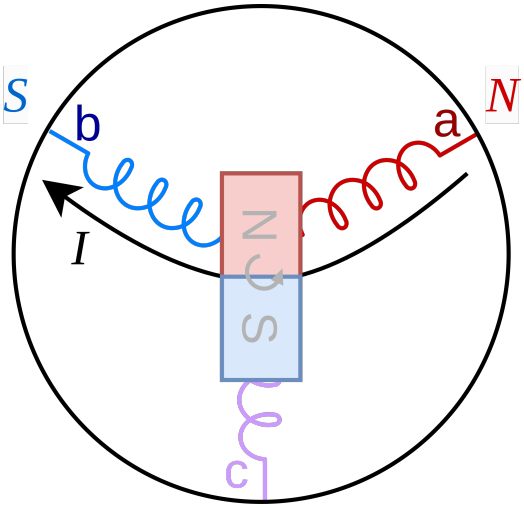

Figure 1.2: Creating a magnetic field around the stator by exciting the A and B phases.

Figure 1.3 Shows the electric motor model connected to the inverter. For the sake of simplicity, the MOSFETs are modelled as switches.

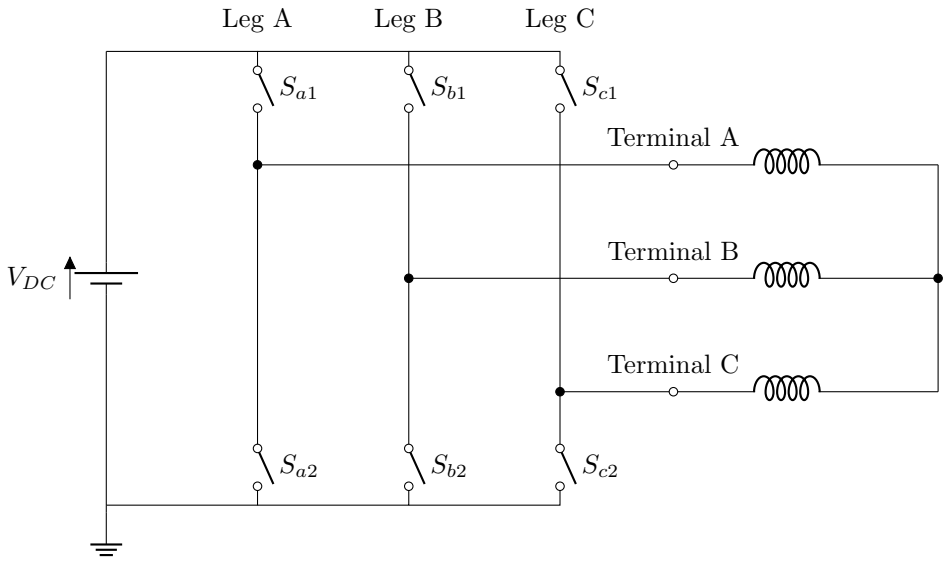

Figure 1.3: The three-phase inverter connected to the motor terminals A, B and C. The motor windings are represented as coils.

It can be seen that each of the motor phases is connected to a leg of the bridge. By manipulating the amount of time that a switch is in the ON or OFF state, the average phase voltages and thus the amount of current through each phase can be regulated. Therefore, by setting the duty cycles of the switches, it is possible to control the phase currents. The way by which these duty cycles are set is dependent on the control algorithm.

1.3.2 General control algorithm structure

The general structure of a control algorithm is depicted in Figure 1.4.

Figure 1.4: A general controller design

The input of the controller is the parameter that must be regulated. For this project, this is the angular velocity. Depending on the control algorithm, this angular velocity can be translated into e.g. a reference torque or flux. The output of the controller consists of the motor commands. As discussed earlier, these are the duty cycles required for each of the MOSFETs in the inverter hardware.

1.3.3 Introduction of existent control algorithms

There are various control algorithms that are suited for the task of controlling a PMSM motor [3]–[5]. In this chapter, a short overview of the algorithms that were considered for this thesis is given. A more detailed description will be given in chapter 5. For each control algorithm, the electrical model of the motor was used.

V/f control

V/f control is an open-loop control algorithm, of which the main operating principle is to set the flux constant by fixing $\frac{V}{f}$. [6]

Field Oriented Control (FOC)

In FOC, the phase currents are transformed to the rotor reference frame. This results in a current component in parallel and a current component orthogonal to the rotor. The orthogonal component (quadrature current) is directly proportional to the torque. FOC controls this quadrature current in order to reach a desired torque [7].

Direct Torque Control (DTC)

Direct torque is similar to FOC. However, it directly controls the motor torque and flux. [8]

Trapezoidal Control

Trapezoidal control splits the motor up into six discrete zones, and drives the terminal voltages based on predetermined values for each zone.[9]

Modified Trapezoidal Modulation (MTM)

Modified trapezoidal modulation works the same as trapezoidal control, however, when trapezoidal control leaves a phase floating, this control algorithm interpolates between voltage values. [10]

Chapter 2

Program of Requirements

All the requirements specified are for the "T motor mn501-s 240kv" PMSM motor[1] configured with a P18×6.1 propeller[2] unless otherwise specified.

2.1 Mandatory Requirements

2.1.1 Planning Requirements

1. The thesis must be completed by the 16$^{\text{th}}$ of June.
2. This subsystem must be fully working by the 30$^{\text{th}}$ of June.

2.1.2 Technical Requirements

3. The control algorithm must be able to reach 140 000 eRPM[1] when the P14×4.8 propeller [11] is attached.
4. The ESC must be able to start with the motor running at any frequency in its operating range.
5. The ESC must be able to continue running after the following abnormalities:
   (a) A sequence of faulty measurements lasting 200 ms at a maximum.
   (b) An unexpected change in speed of at maximum 10 000 eRPM[1] within 10 ms.
   (c) Power interrupts of at maximum 10 s.

2.1.3 Submodule Interoperability Requirements

6. The abnormalities stated for item 5 must be communicated using an 8-bit error code.
7. The ESC must support a maximum MOSFET switching speed of 100 kHz.
8. The control algorithm must have a desired angular velocity in eRPM[1] as input, this value is readable at all times.
9. The control algorithm must work with at maximum three-phase current values and three-phase voltage values as an input.
10. The control algorithm must output the realized angular velocity in eRPM[1], this value must be readable at all times.
11. The control algorithm must output the duty cycles for each gate of the inverter.
12. The control algorithm must prevent instantaneous gate currents over 180 A.

2.2 Trade-off Requirements

13. The control efficiency must at minimum be 70%, this is measured from the output of the inverters to the mechanical output of the motor while the motor is spinning at 50% of its maximum speed.
14. This submodule must be fully integrated with the communication and power submodule by the 16$^{\text{th}}$ of June.

[1] eRPM = 60 × electrical frequency [Hz]

Chapter 3

Design Approach

This chapter explains the approach used for choosing and designing a control algorithm. As discussed in chapter 1, the main challenges lie in converging to a specific algorithm and tuning its parameters in a systematic way.

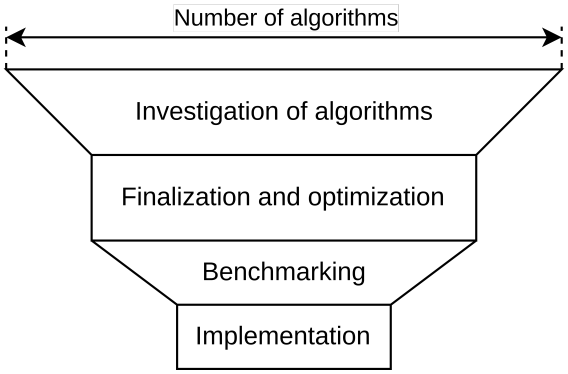

Figure 3.1: This figure graphically shows the design approach used

3.1 Investigation

As depicted in Figure 3.1, the first step is to investigate the following selection of control algorithms: trapezoidal control, modified trapezoidal control, V/f control, FOC, and DTC. The main goal is to determine which algorithms are worth optimizing further and benchmarking. The choice will be made by means of theoretical research and basic simulations.

3.2 Finalization and optimization

The second step is to improve and optimize the implementation of the chosen algorithms in simulation. This, for example, means implementing observers and tuning parameters.

3.3 Performance benchmarking

Once the controllers are in their final state, they are benchmarked and then compared to one another. This is done by means of full system simulations. The algorithm that best meets the requirements is chosen.

3.4 Implementation

The last step is to implement this simulated controller in hardware.

Chapter 4

Simulation setup

As explained in chapter 3, simulations must be run to determine the performance of each of the control algorithms. This chapter explains how the simulator is set up. First, the choice of tools is explained, then a general structure is given. Thereafter each of the components is explained.

4.1 Simulation tools used

Table 4.1: Python or MATLAB

	Python	MATLAB + Simulink
Available resources	Few	Many
Our experience	A lot	Mediocre
Availability to readers	Excellent	Poor/Mediocre

Table 4.1 shows the most important differences between Python and MATLAB. The main advantage of MATLAB is that many of the parts required to build the simulations are already available. It also has a lot of examples written in it. However, we already have a lot of experience programming in Python. And as Python is free and open source, all of our readers will have access to this tool. There also exists a simulation for FOC in Python by employees from TI[12]. Thus, it makes sense to use Python for the simulator.

4.2 General simulation structure

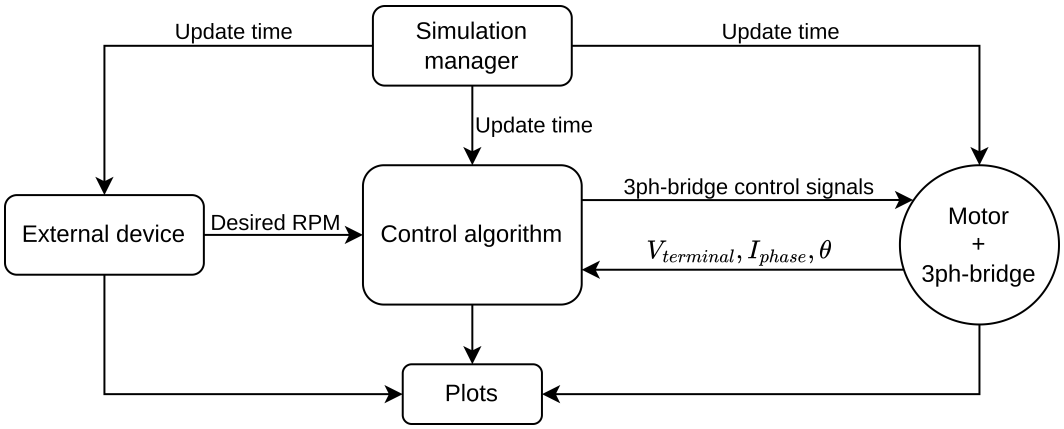

Figure 4.1: The general structure of the simulation.

Figure 4.1 shows the general structure of the simulator. The simulator is split up into five parts. It was chosen to split up the simulator in this way because:

- The simulation script that the simulations in later chapters are based on already made use of a similar model, thus it was quite easy to implement [12].
- It allows the reuse of a majority of the simulator for different control algorithms.

- By splitting the simulation of the control algorithm from that of the rest, it reduces the difficulty during hardware implementation. This part of the simulator code will closely resemble what must be implemented in hardware.

4.3 Simulation manager

The simulation manager runs the update function for each of the other blocks, and it keeps track of the simulated time. It runs 100 cycles of the motor simulator for every cycle of the control algorithm to make the motor simulator approximate a continuous motor.

4.4 External device

The external device determines the desired rotational velocity that is sent to the control algorithm, this replaces what would be a flight controller in a drone. It does this by following a predetermined program that is completely dependent on time. The exact program depends on its configuration.

4.5 Motor model

The motor model is in charge of simulating how the motor reacts to the control algorithm. It does this by means of combining an electrical with a mechanical model.

4.5.1 Electrical model

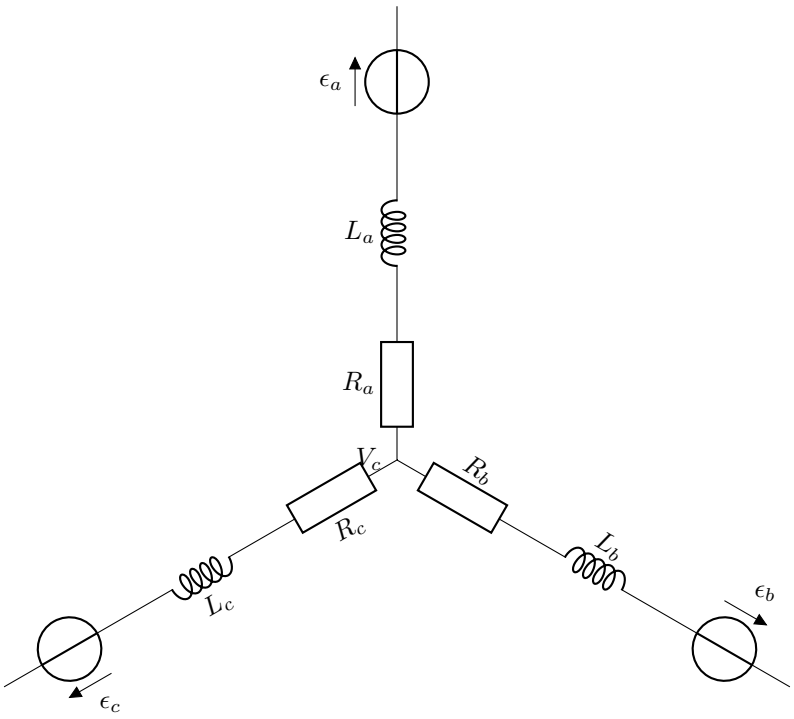

Figure 4.2: The circuit model used for simulating the motor, the model is taken from [12]

Figure 4.2 shows the electrical model used for the motor. For this simulator, it is assumed that $L_a = L_b = L_c$, and that $R_a = R_b = R_c$. The following equations are used to simulate this [12]:

$$V_{L\_phase}[n] = V_{phase}[n] - \epsilon[n-1] - i_{phase}[n-1]R \tag{4.1}$$

$$i_{phase\_intermediate}[n] = i_{phase}[n-1] + \frac{V_{L\_phase}[n]}{L}dt \tag{4.2}$$

Here $V_{L\_phase}[n]$, $V_{phase}[n]$, $\epsilon[n-1]$, $i_{phase}[n-1]$, and R are the voltage over the inductor, the voltage from the neutral point to the terminal, the back-emf voltage, the current and the resistance for each phase respectively. $i_{phase\_intermediate}$ is an intermediate value for the phase current. L is the inductance for each phase, and dt indicates the time each step simulates. L has been determined in Appendix A to be $L = 11.285\,\mu\text{H}$. And the datasheet of

the motor[1] indicates that $R = 0.085\,\Omega$, which has been verified in Appendix B. dt is set to $\frac{1}{f_{control\_algorithm} \cdot 100}$ as becomes clear from section 4.3, specified in the requirements is that $f_{control\_algorithm} = 100\,\text{kHz}$.
To ensure that Kirchhoff's laws holds, the following operation is then performed [12]:

$$i_{avg}[n] = \frac{i_a[n] + i_b[n] + i_c[n]}{3} \tag{4.3}$$

$$i_{phase}[n] = i_{phase\_intermediate}[n] - i_{avg}[n] \tag{4.4}$$

Here $i_{avg}[n]$ is the average current through each phase. $i_a[n]$, $i_b[n]$, $i_c[n]$ are the currents through phases A, B, and C respectively. And $i_{phase}[n]$ is the final calculated current through each phase.

Back-emf

To determine the back-emf, the following equation is used [12]:

$$\epsilon[n] = \frac{1}{\sqrt{3}} \omega[n] K_t \sin(\theta[n]) \tag{4.5}$$

Where $\omega[n]$ is the rotational velocity in $\text{rad}\,\text{s}^{-1}$. K_t is the torque constant in $\text{V}\,\text{rad}\,\text{s}^{-1}$, which is taken from the datasheet of the motor [1]. $\theta[n]$ Is the electrical phase angle of the rotor in rad. The $\frac{1}{\sqrt{3}}$ factor was taken to convert the back-emf from a DC peak-to-peak value to a phase amplitude. Please note that the phase angles of the three phases will differ by $\frac{2}{3}\pi$.

4.5.2 Mechanical model

For the mechanical model, the following equations were used [12]:

$$T[n] = K_T(i_a[n]\sin(\theta_a[n]) + i_b[n]\sin(\theta_b[n]) + i_c[n]\sin(\theta_c[n])) \tag{4.6}$$

$$\omega[n] = \omega[n-1] + dt \cdot \frac{T - \omega[n-1] \cdot F}{I} \tag{4.7}$$

$$\theta[n] = \theta[n-1] + dt \cdot \omega[n] \tag{4.8}$$

Here $T[n]$ is the torque. K_T is the torque constant of the motor. $i_x[n]$ is the current through phase x. $\theta_x[n]$ is the relative angle of the rotor to phase x. $\omega[n]$ is the rotational velocity. F is the coefficient of friction. I is the rotational inertia. $\theta[n]$ is the rotor angle. And dt is the time delta of the simulation step.

Inertia

The inertia of the propeller is estimated by assuming the propeller is a rad. Thus $I_{prop} = \frac{1}{12} m_b l_b^2$, using numbers from the datasheet [2] it can be calculated that $I_{prop\_large} = 5.49 \times 10^{-4}\,\text{kgm}^2$. Assuming that the rotor of the motor can be approximated as a hollow cylinder and that it has $\frac{1}{3}$ of the mass of the motor, $I_{rotor} = 4.38 \times 10^{-5}\,\text{kgm}^2$. Thus $I_{total\_large} = 5.93 \times 10^{-4}\,\text{kgm}^2$.
For the requirements of maximum rotational velocity, a smaller propeller was used [11]. Doing the same calculations as above for this propeller results in $I_{total\_small} = 2.02 \times 10^{-4}\,\text{kgm}^2$.

Friction

The motor datasheet [1] gives performance numbers for the small propeller. The current can be used to calculate the torque delivered using the following formula $T = ik_t = 0.74\,\text{N m}$. This torque can then be divided by the speed reached to find the coefficient of friction: $F_{small} = \frac{T}{\omega} = 7.13 \times 10^{-4}\,\text{N m s rad}^{-1}$.
The motor datasheet only gives these numbers for a similar motor with a different torque constant. As this motor has approximately the same mechanical design, these numbers are still quite accurate. The formulas result in $F_{large} = 1.52 \times 10^{-3}\,\text{N m s rad}^{-1}$ for this propeller.

4.5.3 Floating terminals

The motor model given by [12] can not simulate floating terminals. However, to correctly be able to implement zero-crossing detection for trapezoidal motor control, this feature is required. More on this can be found in section 5.4. It is also required by the sensing start mechanism proposed in section 6.1.2.
Unlike driven terminals, the voltages of floating terminals are not fixed by the control algorithm. Thus, these must be calculated by means of the other terminal voltages and the back-emf. Assuming the model from Figure 4.2,

that no current flows through the floating terminal, and neglecting the instantaneous voltage peak generated by this, for a single floating terminal, the following equation can be used:

$$V_c = \frac{1}{2}V_{terminal1} + \frac{1}{2}V_{terminal2} - \frac{1}{2}\epsilon_1 - \frac{1}{2}\epsilon_2 \tag{4.9}$$

$$V_{terminal3} = V_c + \epsilon_3 \tag{4.10}$$

Here ϵ is back-emf voltage, and terminal 3 is floating while terminals 1 and 2 are driven.
For two floating terminals using the same assumptions, the following equations are used:

$$V_c = V_{terminal1} - \epsilon_1 \tag{4.11}$$

$$V_{terminal2} = V_c + \epsilon_2 \tag{4.12}$$

$$V_{terminal3} = V_c + \epsilon_3 \tag{4.13}$$

Here terminals 2 and 3 are floating while terminal 1 is driven.
As non of the controllers simulated for this report ever set all three terminals floating, this is not simulated, if this ever occurs, an error is thrown instead.

4.5.4 Forward voltage

The equations given in subsection 4.5.3 allow for phase voltages levels outside the DC bus voltage range. However, due to the internal diodes within the MOSFETs of the inverter, this is not possible in reality. These diodes are simulated as it was noticed that for trapezoidal control voltage levels under zero were often reached without this. Thus, to make sure the simulation results are accurate, it was decided to implement this part of the drive circuit. For this simulator, the diodes are assumed to be ideal with zero forward voltage.
The diodes are simulated by limiting the terminal voltage to a range between 0 and the DC supply voltage. When the voltage is limited, it can be seen as driven, thus the current is not zero any longer. The power delivered back to the DC bus is calculated by means of the following equation:

$$P_{diodes} = |(V_a - V_{a\_limited}) \cdot i_a| + |(V_b - V_{b\_limited}) \cdot i_b| + |(V_c - V_{c\_limited}) \cdot i_c| \tag{4.14}$$

Here V_a, V_b, and V_c are the phase voltages before the voltage limiting is performed of phases a, b, and c respectively. And $V_{a\_limited}$, $V_{b\_limited}$, and $V_{c\_limited}$ are the phase voltages after the voltage limiting is performed of phases a, b, and c respectively.

4.5.5 Energy

The simulator also calculates the energy used by each controller. It makes use of the following equations:

$$E_L(t) = \frac{1}{2}L(i_a(t)^2 + i_b(t)^2 + i_c(t)^2) \tag{4.15}$$

$$E_R(t) = \int_0^t R(i_a^2 + i_b^2 + i_c^2)(\tau)\, d\tau \tag{4.16}$$

$$E_{mechanical}(t) = \int_0^t T(\tau)\omega(\tau)\, d\tau \tag{4.17}$$

$$E_{electrical\_total}(t) = E_L(t) + E_R(t) + E_{mechanical}(t) - \int_0^t P_{diodes}(\tau)\, d\tau \tag{4.18}$$

Here i_a, i_b, i_c, are the phase currents through phases a, b, and c respectively. $E_L(t)$ is the total energy in all three inductors. $P_{diodes}(t)$ is as explained in subsection 4.5.4. $E_R(t)$ is the total energy dissipated by the resistors. $E_{mechanical}(t)$ is the total mechanical energy that is delivered by the motor, which is assumed to be equal to the electrical energy consumed by the back-emf. Thus, $E_{electrical\_total}(t)$ is the total electrical energy that is delivered by the DC bus.

Chapter 5

Simulation and investigation of algorithms

This chapter looks into each of the proposed control algorithms to determine if they could be used in a setup that conforms with the project requirements. If they may be able to, they will be investigated in the following chapters. Otherwise, the control algorithm will not be investigated further.
The full simulation results for this chapter can be found in the appendices.

5.1 V/f control

A V/f controller assumes quasi-steady state. With this assumption, one can fix the air gap flux by setting $\frac{V}{f}$ constant, assuming that the motor resistance is negligible. This is exactly what is done with the V/f controller depicted in Figure 5.1. As can be seen, the system works fully open-loop. This simple idea is depicted in Figure 5.2 by the linear region. However, as the voltage is limited due to the construction of the motor and control hardware, this is not maintainable at higher speeds, thus only the frequency is varied. This means that the air gap flux weakens in what is indicated as the field weakening region in the figure. [6], [13]
The resistance of the motor must be compensated for at a lower rotational velocity, as it is not negligible any longer, to achieve this, the voltage is pushed up in what is depicted as the voltage drop compensation region. [6]
The waveform generator converts the voltage and frequency values into a waveform. It does this by integrating the frequency and then using this to generate three sinusoidal waveforms with a 120-degree phase difference. The voltage value then sets the amplitude. [6]
Also, a closed-loop form of this control algorithm exists which is used to compensate for motor slip. But as a PMSM motor makes use of permanent magnets, there is no slip, thus, the closed-loop form is not useful for this motor type. [6], [13]

5.1.1 Simulation

To determine if open-loop V/f control could comply with the requirements, a simulation is run. For this simulation, the design structure as given in Figure 5.1 has been implemented.

V/f curve

As explained above, Figure 5.2 gives the V/f curve for a V/f controller. For this simulation, the voltage is set to a minimum of 5 volts to compensate for stator losses at low speeds, this value was determined by means of trial and error to be stable. The voltage is set to a maximum of 25 volts, as the DC bus can not deliver higher voltages.

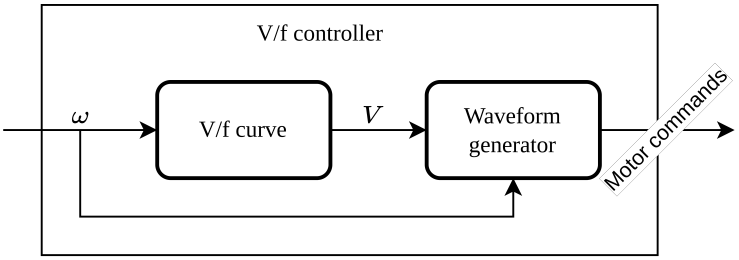

Figure 5.1: A basic open-loop V/f controller design [6]

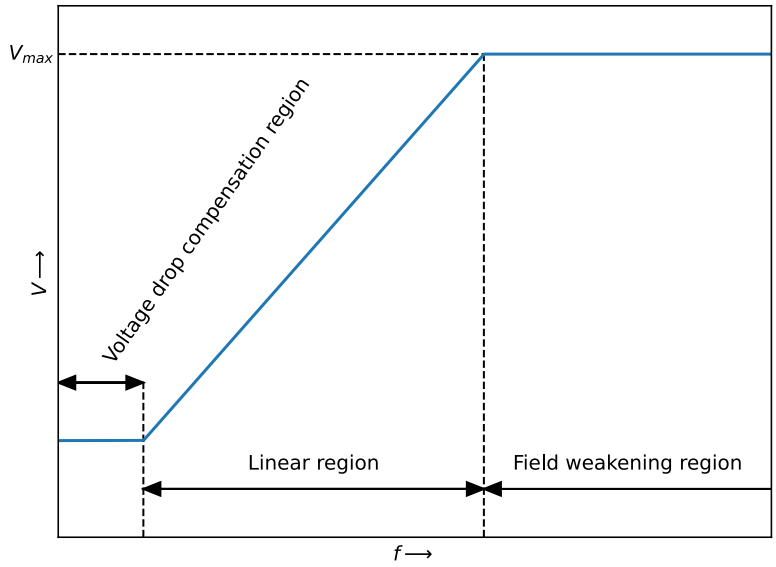

Figure 5.2: A typical voltage over frequency curve for a V/f controller [6]

For this simulation $\frac{V}{f}$ is fixed to $3.6 \times 10^{-3}\,\mathrm{V\,s\,rad^{-1}}$, which was determined by trial and error to generate reasonable values for the voltage and currents through the motor.

Waveform generator

During each time step, the new voltage values for the motor are calculated by the waveform generator. The following calculations are performed:

$$\theta[n] = \theta[n-1] + dt \cdot \omega[n] \tag{5.1}$$

$$V_{a\_normalized}[n] = \frac{1}{2} + \frac{V[n]}{25} \cdot \sin(\theta[n]) \tag{5.2}$$

$$V_{b\_normalized}[n] = \frac{1}{2} + \frac{V[n]}{25} \cdot \sin(\theta[n] - \frac{2\pi}{3}) \tag{5.3}$$

$$V_{c\_normalized}[n] = \frac{1}{2} + \frac{V[n]}{25} \cdot \sin(\theta[n] + \frac{2\pi}{3}) \tag{5.4}$$

Here $\omega[n]$ in the desired frequency in $\mathrm{rad\,s^{-1}}$, $\theta[n]$ is the calculated stator angle in rad, $V_{a\_normalized}[n]$, $V_{b\_normalized}[n]$, $V_{c\_normalized}[n]$ are the normalized voltages applied to phases A, B, and C respectively. Which perfectly aligns with the duty cycle required to generate the required voltages. $V[n]$ is the voltage calculated as explained in section 5.1.1, it is divided by 25 to normalize it.
These equations generate three sinusoidal waveforms that are 120 degrees out of phase from one another. They are varied around $\frac{1}{2}$ as the inverter can not generate negative voltages.

Results

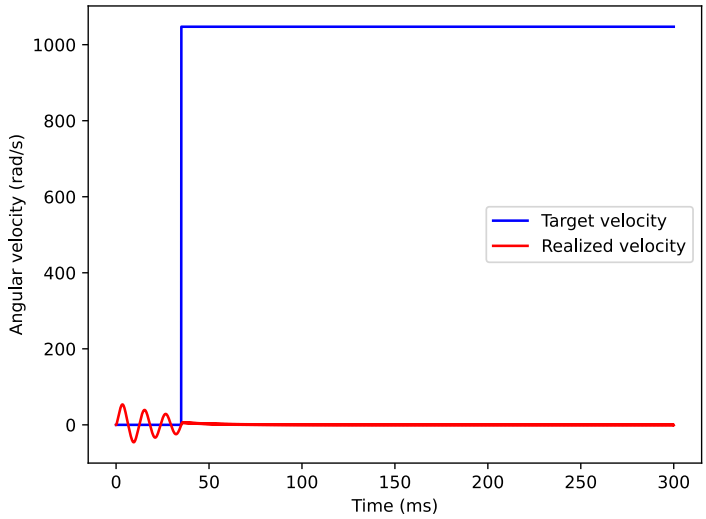

Figure 5.3: V/f control simulation following the desired angular velocity.

Figure 5.3 shows the simulation results of this V/f controller. As can be seen, it is not able to follow abrupt changes in velocity. Thus, this design will not work for this project.

Rate limiter

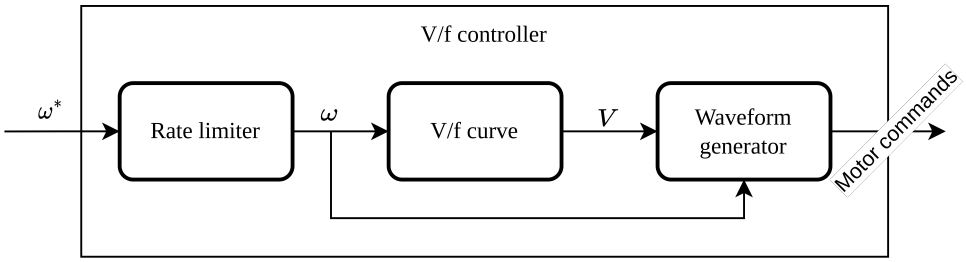

Figure 5.4: V/f control schematic with rate limiter.

To prevent abrupt changes in speed from destabilizing the controller, a simple rate limiter can be added as shown in Figure 5.4. Trial and error has shown that a rate limit of $100 \times 10^{-3}\,\mathrm{rad/s^2}$ creates a stable, but relatively fast responding, system.

Results with rate limiter

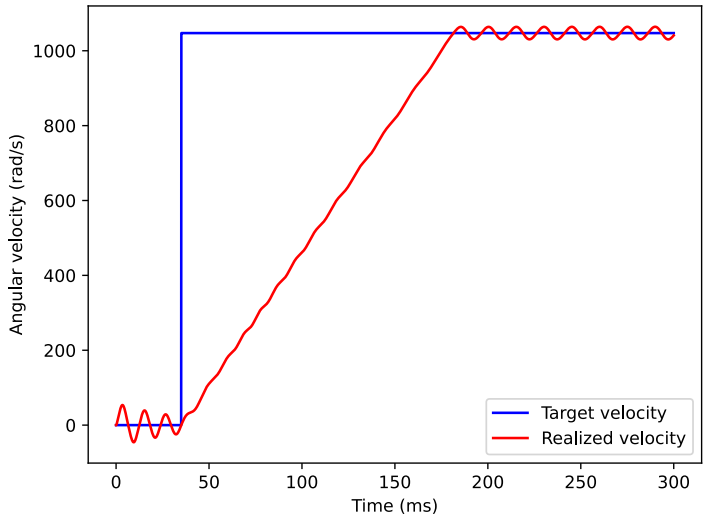

Figure 5.5: V/f control simulation following the desired rotational velocity with a rate limiter.

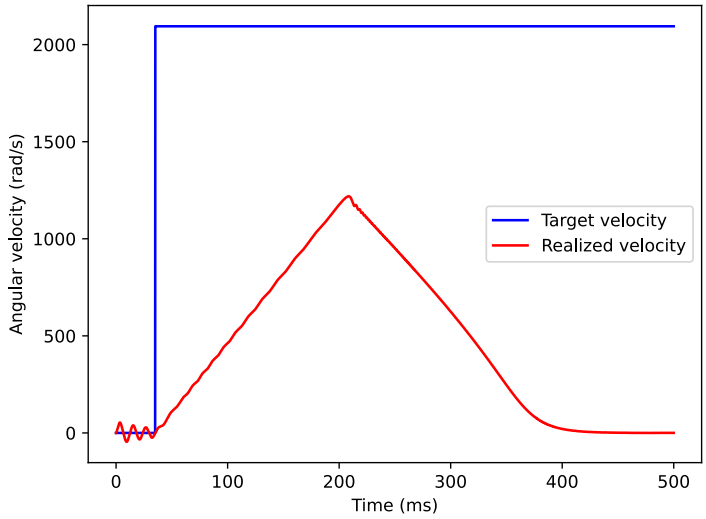

Figure 5.6: V/f control simulation following the desired rotational velocity with a rate limiter. With a larger desired speed.

Figure 5.5 shows the results from a simulation with this implemented. As can be seen, it does solve this issue. However, if a torque is desired that cannot be reached, this controller will not be able to follow, which results in an abrupt drop in speed, this is shown in Figure 5.6. When a larger load is attached this also becomes problematic for lower speeds, thus, this controller is not considered further.

5.2 Field Oriented Control (FOC)

5.2.1 Global principle

As was explained in chapter 1 the amount of current flowing through each motor phase can be regulated by controlling the duty cycles of the MOSFETs in the inverter. The current flowing through the motor phases creates a magnetic field inside the motor which exerts a torque on the rotor. As has been illustrated in Figure 1.2 in chapter 1, the rotor in the electrical motor model is equivalent to the rotating magnetic field inside the motor. It is, however, only the magnetic field component that is orthogonal to the rotor that contributes to the torque. A

current vector parallel to the rotor would create a magnetic field that is also parallel to the rotor. Depending on the polarity of this magnetic field, the flux inside the motor would be minimized or maximized, while the torque exerted on the rotor would remain zero. This means that the torque per ampere ratio in such a case is zero. FOC aims to maximize this ratio by ensuring that the vector sum of the currents flowing through the three motor phases is always orthogonal to the rotor. The current component orthogonal to the motor is referred to as the quadrature current and the current component parallel to the rotor is the direct current. The torque exerted on the rotor is directly proportional to the quadrature current:

$$\tau = I_q \cdot K_\tau \tag{5.5}$$

Here, I_q is the quadrature current in A, τ is the torque in N m and K_τ is the torque constant in $\mathrm{N\,m\,A^{-1}}$. The torque constant can be calculated from the motor speed constant that is available in the datasheet of the "T motor mn501-s 240kv" motor [1] in the following way:

$$K_{\tau,SI} = \frac{1}{K_{v,SI}} = \frac{60}{2\pi \cdot K_{v,RPM}} \tag{5.6}$$

Thus, by controlling the magnitude of the quadrature current, it is possible to follow a reference torque. This sums up the purpose of FOC: it aims to keep the direct current at zero and to control the magnitude of the quadrature current in such a way, that a reference torque is reached.

5.2.2 Control Scheme

Figure 5.7 shows the control scheme that has been used for the first simulation of FOC. The code that has been used is heavily based on [12]. First, a reference quadrature and direct current are set. These reference currents are calculated based on the target torque set by the user.

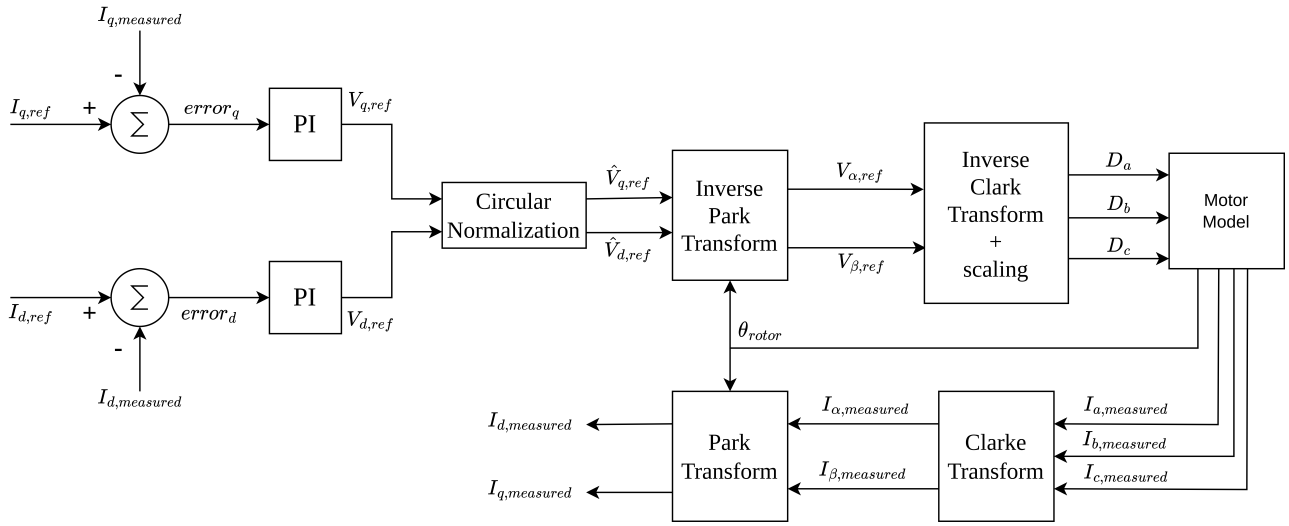

Figure 5.7: FOC scheme as used for the simulation. Please note that the measured and quadrature current from the motor are the same measured direct and quadrature current that are fed to the PI controller. This has not been explicitly shown in the figure in favor of legibility.

The direct and quadrature currents, however, cannot be measured directly. The inverter circuit shown in Figure 1.3 is equipped with sensors that can measure the phase currents. In order for FOC to function, these need to be transformed to the rotor reference frame. The transformation is done in two steps. First, the three phase currents are transformed to a two-phase reference frame. This is the $\alpha\beta$-frame, as depicted in Figure 5.8.

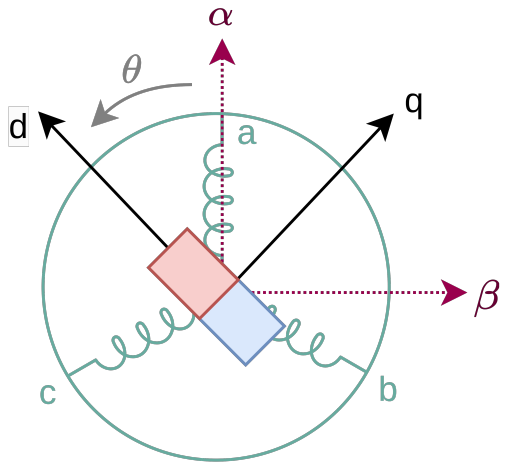

Figure 5.8: The three-phase (abc), two-phase ($\alpha\beta$) and rotor reference (dq) frames with respect to the electrical motor model.

The transformation to the $\alpha\beta$-frame is done by applying the Clarke transform, as shown in Equation 5.7.

$$\begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix} = c \cdot T_C \cdot i_{abc}(t) = c \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} \tag{5.7}$$

Here, $i_{\alpha,\beta}$ are the two-phase currents in A, $i_{a,b,c}$ are the three-phase currents in A, and c is a scaling constant. When this scaling constant is unity, the amplitudes of the phase currents are not preserved across the transform. This means that when, for example, 1 A is run through phase a, this will not translate to 1 A on the α-axis, but to $\frac{2}{3}$A. In order to preserve the current magnitudes across the transform, scaling constant c should be set to $\frac{2}{3}$. In such a way, the amplitude invariant Clarke transform is obtained. It should be noted that Equation 5.7 assumes a balanced system, which means that the following holds:

$$i_a + i_b + i_c = 0 \tag{5.8}$$

As the electrical motor model used for the simulations is Y-wound, Equation 5.8 holds because of Kirchhoff Current Law. The $\alpha\beta$-currents are then transformed to the dq reference frame, as shown in Figure 5.8. This dq reference frame is the rotor frame. The transformation is done by applying the Park transform:

$$\begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix} = T_P \cdot i_{abc}(t) = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix} \tag{5.9}$$

Here, $i_{d,q}$ are the direct and quadrature currents and θ is the rotor electrical angle in $\mathrm{rad\,s^{-1}}$, as shown in Figure 5.8. For the initial simulations, a sensored system has been assumed. This means that the rotor angle is directly measurable from the motor. In reality, however, this will not be the case. Thus, if FOC were to be investigated further, a solution for sensorless angle estimation would need to be found.
Figure 5.7 shows how the differences between the measured and reference direct and quadrature currents are calculated and fed to PI controllers. The outputs of the PI controllers are the quadrature and direct voltages. The magnitude of the vector sum of $V_{q,ref}$ and $V_{d,ref}$ is normalized in such a way that it does not exceed unity, and thus is bounded by a circle of unity radius. The radius of the circle corresponds to the maximum duty cycle, and thus the maximum amplitude of the terminal voltage that can be achieved. In this case, the maximum possible amplitude of the phase voltage is $\frac{V_{DC}}{2}$, where V_{DC} is the DC bus voltage. The inverse Park and inverse amplitude invariant Clarke transform bring the reference voltages back to the three-phase stator reference frame. The inverse Park and Clarke transformations are shown in Equation 5.10 and Equation 5.10, respectively.

$$\begin{bmatrix} v_\alpha(t) \\ v_\beta(t) \end{bmatrix} = T_P^{-1} \cdot v_{dq}(t) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} v_d(t) \\ v_q(t) \end{bmatrix} \tag{5.10}$$

$$\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} = \frac{3}{2} \cdot T_C^{-1} \cdot v_{\alpha\beta}(t) = \frac{3}{2} \begin{bmatrix} \frac{2}{3} & 0 \\ -\frac{1}{3} & \frac{\sqrt{3}}{3} \\ -\frac{1}{3} & -\frac{\sqrt{3}}{3} \end{bmatrix} \begin{bmatrix} v_\alpha(t) \\ v_\beta(t) \end{bmatrix} \tag{5.11}$$

Note that $V_{q,ref}$ and $V_{d,ref}$ will rotate along the path of the unity circle at the same angular velocity as the rotor in the electrical motor model. Directly translating this to three-phase voltages will result into sinusoids varying from -1 to 1, which corresponds to sinusoidal terminal voltages varying from $-\frac{V_{DC}}{2}$ to $\frac{V_{DC}}{2}$. The inverse Clark transform has been scaled in order to ensure that the output duty cycles vary from 0 to 1, instead of from -1 to 1. This way, the terminal voltages are sinusoids that vary from 0 to V_{DC} and have an amplitude of $\frac{V_{DC}}{2}$. This PWM modulation scheme is therefore named Sinusoidal Pulse Width Modulation (SPWM), and will result into phase voltages that vary from $-\frac{V_{DC}}{2}$ to $\frac{V_{DC}}{2}$. When the duty cycles are fed to the motor, the motor phase currents are measured, translated to stator reference frame and used to calculate the error signals which are fed to the aforementioned PI controllers.

5.2.3 Simulation Results

Figure 5.9 and Figure 5.10 show the simulation results for the FOC control scheme discussed in the previous section. Figure 5.9 shows that the phase voltages are sinusoids varying from $-\frac{V_{DC}}{2}$ to $\frac{V_{DC}}{2}$ ($-25\,\text{V}$ to $25\,\text{V}$), which aligns with what should be expected from the PWM modulation scheme that was used for this simulation. Figure 5.10 shows that FOC is able to make the motor spin at a speed that is higher than the maximum desired angular velocity. It can be seen that FOC is not yet able to follow a requested speed. This is due to the fact that a speed controller is not yet implemented. With the implemented PI controller, it is only possible to set a desired torque. In order to properly assess the potential of FOC, this PI controller has been systematically tuned. The method, however, is out of scope for this chapter and will be described in detail in chapter 6. With these simulation result, it was concluded that FOC is worthy of further investigation. In further research, a solution needs to be found for the estimation of the rotor angle. Also, a speed controller will need to be designed and tuned. Lastly, the PWM modulation scheme will be looked into. The modulation scheme that has been used for this simulation only allows for phase voltages varying from $-\frac{V_{DC}}{2}$ to $\frac{V_{DC}}{2}$. However, there are other modulation schemes that allow for a better utilization of the DC bus voltage. These will be further investigated in chapter 6.

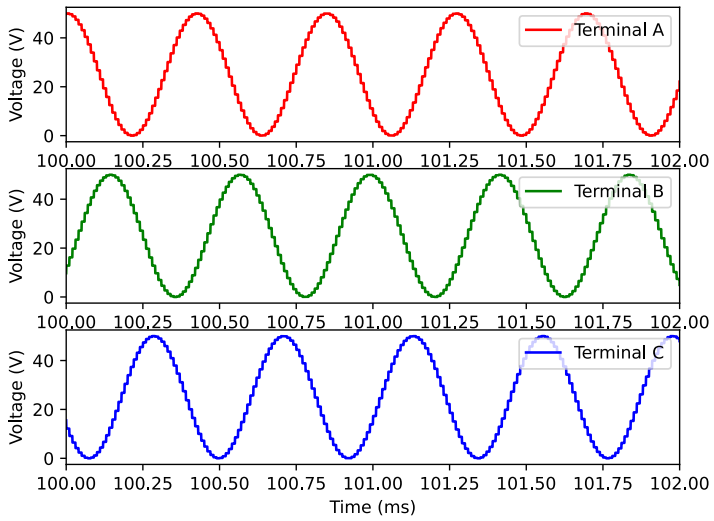

Figure 5.9: The phase voltages from a simulation, as can be seen the phase voltages are sinusoidal.

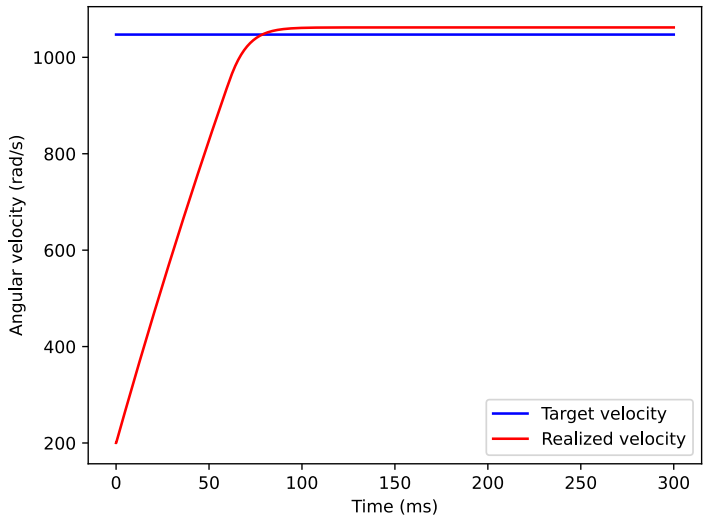

Figure 5.10: A plot of the target velocity and the realized velocity, as can be seen FOC is able to reach the required maximum speed.

5.3 Direct torque control

Direct torque control makes use of the voltages provided to the motor and measured currents to estimate the torque and flux. For the flux, a two-state hysteresis controller is used, and for the torque, a three-state hysteresis controller is used. The outputs of these controllers and the direction of the flux are used by a switching table with pre-determined voltage levels to determine the required output voltage.
Direct torque control is quite simple and does not require a lot of processing power. However, it does suffer from torque ripple due to the use of hysteresis controllers. [8], [14]–[16], and tuning takes a lot of time. Thus, due to time considerations, it is not worth considering further. [15]

5.4 Trapezoidal control

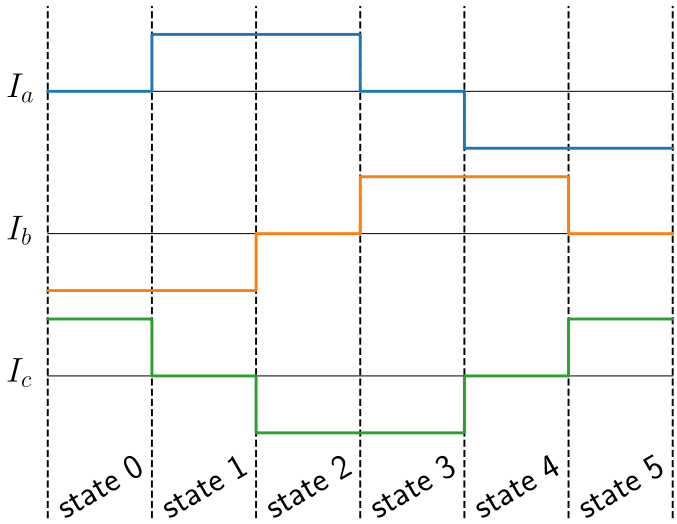

Figure 5.11: States of trapezoidal control [17]

Trapezoidal control makes use of a six-step cycle to control the motor, as shown in Figure 5.11. As can be seen, only two terminals are driven concurrently, which means that voltage sensing can be used to determine the rotor position of the motor. And as the algorithm only uses 6 discreet steps, the rotor position must only be accurate at six discrete points, simplifying the sensing system [17], [18]. Because this control algorithm does not need rapid changes in the voltage levels, a buck converter could be placed before the three-phase bridge inverter to reduce

switching speeds. [10]
However, trapezoidal control is commonly used for BLDC motors, which is a type of motor that is optimized for using a trapezoidal shaped supply currents, as seen in the image. Using this type of modulation for a PMSM motor can cause torque ripples, which cause vibrations. [18]–[22]

5.4.1 Simulation

It was opted to run a simulation to determine the effectiveness of trapezoidal control with the PMSM motor specified in chapter 2, as not many sources are available on the performance of this algorithm to drive PMSM motors.

General structure

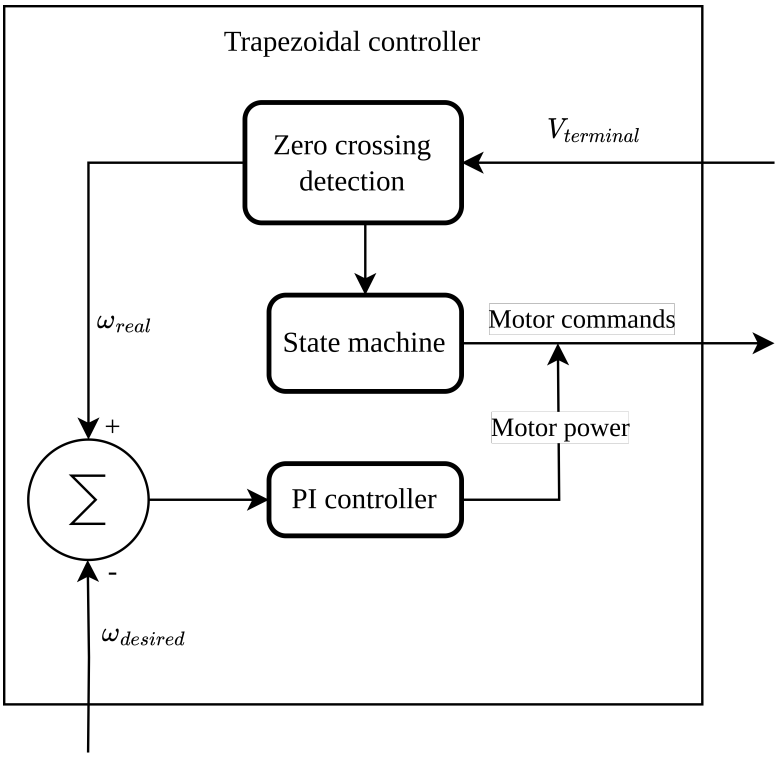

Figure 5.12: Design of the simulated trapezoidal controller [9]

Figure 5.12 shows a general structure of a trapezoidal controller. Figure 5.12 shows the overall design of the trapezoidal controller that is implemented for this simulation. It closely resembles the implementation by [9]. As shown, terminal voltages are first converted to crossing times by the zero crossing detection block, more on this in section 5.4.1. These values are then sent to the state machine, which converts them to motor commands.
The zero crossing detection block also outputs a measured rotational velocity, this is compared to the desired rotational velocity and forwarded to a PI controller. The PI controller determines the required motor power command. The motor power is combined with the motor commands to determine the duty cycle for each phase.

Zero crossing detection

As shown in Figure 5.13, there are six states. Each state corresponds with a section of the motor rotation. To be able to determine when the state controller must switch to the next state, a method must be implemented to sense the rotation of the rotor. For this, zero crossing detection is commonly implemented. As can be seen in the schematic of Figure 5.13, in a PMSM motor a permanent magnet moves past a coil. While moving, a back-electromotive force is generated. If the north pole passes the coil, a rising back-electromotive force is generated, if the south pole passes, this voltage is falling. One interesting characteristic is that this voltage passes through zero when the magnet and the coil are aligned. The zero crossing block detects this to determine exactly when this occurs. [9], [18]

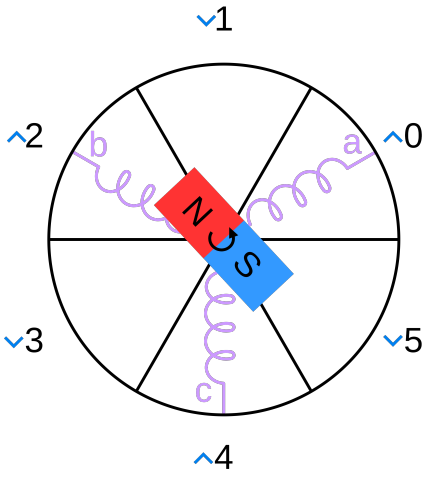

Figure 5.13: Simplified motor model and states. The numbers indicate the state numbers of the sections. The inductors indicate the relative position of the inductors in the motor. The blue arrows indicate if the back-emf is rising or falling during a state. And the arrow indicated in which direction the magnet is rotating.

The simulator first calculates the back-emf using the following equation:

$$V_c = \frac{V_{terminal1} + V_{terminal2} + V_{terminal3}}{3} \tag{5.12}$$

$$\epsilon = V_{terminal1} - V_c \tag{5.13}$$

Here V_c is the center voltage of the motor, as indicated in Figure 4.2. $V_{terminal1}$ is the terminal that is being sensed. $V_{terminal2}$ and $V_{terminal3}$ are the two terminals that are driven by the trapezoidal controller, thus, these are simply equal to the output of the controller. ϵ is the difference between the voltage of the floating terminal and the center voltage, as there is no current flowing through this phase, this difference must be equal to the back-emf. The module then determines if a zero crossing has occurred by comparing the sign of the back-emf of the current cycle with that of the previous cycle. If they are different, it means a zero crossing must have occurred.
The time of the zero crossing is then used to determine when the system should switch to the next state. At constant angular velocity, the time remaining for the current state, after a zero crossing, is half that of the time between the last two zero crossings. This is used by the state machine to determine when to switch states. [9]

State machine

Figure 5.13 shows the six states of trapezoidal control. At the rotation depicted, coil B is sensing the back-emf. To make this possible, the terminal should be left floating. Thus, both of the MOSFETs for this phase are left open. In its current position, the coil at phase B must repel the south pole and attract the north pole as the north pole should move towards it. A must do the exact opposite, as the north pole should move away from it. [17]
As the same operating principle also holds for other orientations, that the waveform must look as shown in Figure 5.11.

PI controller

The PI controller is placed for it to remove the error between the realized angular velocity and the desired rotational velocity. This is used to actually control the speed of the motor. [9]
As this chapter is meant to determine if it is useful to further investigate this algorithm, it was chosen to tune it using trial and error as it is a simple and quick method. The disadvantage is that it means that the simulations will not show the maximum acceleration that this algorithm can achieve, or the velocity may show some overshoot. If it turns out that the algorithm is worth considering further, it will be tuned in a more accurate way in the next chapter.

Results

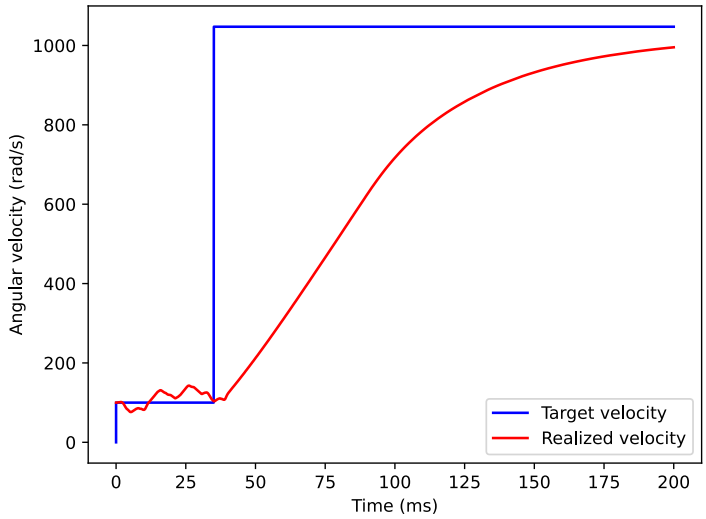

Figure 5.14: Trapezoidal simulation following the desired rotational velocity.

As can be seen in Figure 5.14, trapezoidal simulation is able to follow velocity targets, and is able to almost reach the maximum desired rotational velocity. Thus, the simulation indicates that it is most likely possible to reach the desired requirements using trapezoidal control. Which means that this control algorithm in later chapters.

5.5 Modified trapezoidal control

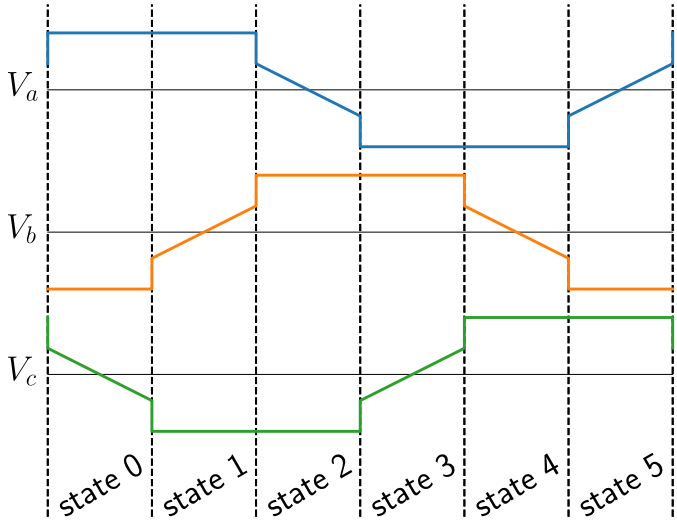

Figure 5.15: States of trapezoidal control

Modified Trapezoidal Modulation (MTM) is a relatively new control algorithm. Its implementation is quite similar to that of the trapezoidal control algorithm, however, where trapezoidal control would leave one terminal floating, MTM drives the terminal with a linearly interpolated voltage between two points. This is shown in Figure 5.15. Compared to the sinusoidal control algorithms proposed below, a lower switching speed is needed if a buck converted is used for speed control [10]. This increases efficiency and decreases torque ripple with the cost of added complexity as a more complicated rotor position sensing method must be implemented. Unfortunately, designing an observer that works with this controller is out of the scope of this project. Thus, it will not be further considered.

5.5.1 Simulations

Even though, this algorithm will not be considered for further research. Simulations which makes use of an encoder have been run to give an indication of the performance that could be achieved if one were to want to research this algorithm further. These simulations can be found in Appendix Appendix H.

5.6 Conclusion

Both trapezoidal control and field oriented control show promising results. Modified trapezoidal modulation, and direct torque control would also be quite promising if more time were allocated for this project. However, to keep the scope limited and to reduce the risk of no result, these are not further considered. V/f control does not guarantee stability, and therefore is unfit for the task of controlling the motor.

Chapter 6

Optimizing and Benchmarking

Chapter 5 comes to the conclusion that trapezoidal control and field oriented control are worth investigating further, which is done in this chapter. First, the designs for the control algorithms are brought to a state in which it makes sense to compare them. Then, these final designs are benchmarked and compared to one another.

6.1 Trapezoidal control

In section 5.4 trapezoidal control was tested by means of simulations using a preliminary controller. This section continues onto the controller presented there.

6.1.1 Zero-crossing by means of integration

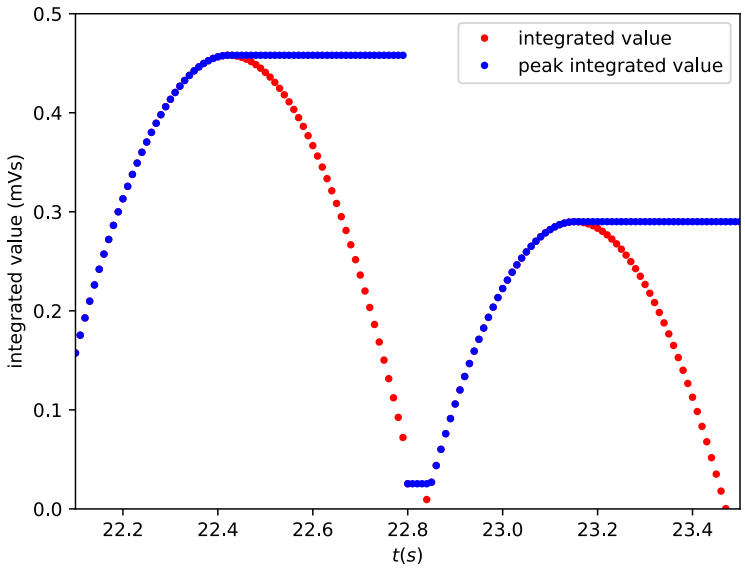

Figure 6.1: An example of zero crossing by means of integration. This figure comes from a simulation.

To detect zero-crossings, the earlier design simply compares the sign of the calculated back-emf of the current cycle with that of the previous cycle. This works quite well in simulation, however, it is not an ideal system when noise is present.
Another method to find the zero-crossings is to make use of integration. It is very similar to the system described in section 5.4.1, however, it differs in one important way. Instead of comparing the sign of the current back-emf to that of the one calculated in the previous cycle, the value of the back-emf is integrated. The peak of this integrated value then corresponds to the where the back-emf is zero. Figure 6.1 shows this system in action.
As the state change occurs in between zero crossings. Unlike the previous method, this method is able to update the time of the zero-crossing when a new, higher, peak is detected.
As the noise is integrated using this method, it partially cancels itself out, thus, its influence is reduced. This means that, this method is more resilient to noise.

6.1.2 V/f for low speeds

Another issue is that zero crossing detection does not work well at low speeds, as the hardware is not able to sense the low back-emf voltages. Thus, it was chosen to implement an open loop V/f controller to start the motor.

Starting procedure

In the requirements it is stated that the motor must be able to start running from any frequency in its operating range. To do this, the back-emf can be measured before it is chosen which algorithm is used. It is assumed that a back-emf of 2 volts is enough to measure accurate zero-crossings. To determine how long has to be measured, the following formula can be used:

$$\omega = \epsilon k_v \tag{6.1}$$

Thus, for 2 volts, $\omega = 16\pi \mathrm{rad\,s^{-1}}$. Thus, $\omega_{el} = 704\,\mathrm{rad\,s^{-1}}$. If the system measures the voltages in all three windings, one peak can be measured for every sixth of an electrical cycle. Thus, the system must at least measure $\frac{2\pi}{\omega_{el} \cdot 6} = 1.49 \times 10^{-3}\,\mathrm{s}$. If no peak over 2 volts was detected on any of the phases, this means, it will never be detected. And thus, V/f control must be used. Otherwise, the system can boot up into trapezoidal control.

Switching to trapezoidal control

When the system is in V/f control mode, it must switch to trapezoidal control once the rotational velocity is high enough. To do this, the rotational velocity after rate limiting is used. To avoid the system from switching more than once, a higher rotational velocity than the threshold as given in section 6.1.2 must be taken. To be on the safe side, ω_{el} is taken to be $1200\,\mathrm{rad\,s^{-1}}$.
During a switch, the current angle in the waveform generator is used to determine the state of the trapezoidal controller. The rotational velocity is calculated back to zero-crossing times, and the voltage value of the V/f controller is converted into an equivalent I value for the speed controller of the trapezoidal controller.

Switching from trapezoidal control

If a rotational velocity under the threshold velocity is desired, the system must be able to switch from trapezoidal control back to V/f control. To determine when the system must switch, the speed estimated by zero crossing detection is used. The transition is always performed at the exact moment that the trapezoidal controller would switch states, as the rotor rotation is known at this time. This motor rotation is forwarded to the waveform generator of the V/f controller together with the last measured angular velocity of the trapezoidal controller.

Tuning the V/f controller

As it is expected that the V/f controller is mainly used for a few milliseconds during startup, optimizing it will not result in much better performance of the overall system. Thus, it was chosen to use the same values as had been used in section 5.1.1. However, as it is not especially important that the controller closely follows the desired rotational velocity, the rate limiter is tuned down to a maximum rate of $100 \times 10^3\,\mathrm{rad/s^2}$.

6.1.3 Tuning the speed controller

In the earlier design, the controller was tuned by hand, as it was mainly important to get an indication of the performance that can be achieved. For benchmarking, however, a more optimal design is required. Thus, a more systematic approach must be used. This can be done by means of the transfer function. For this, the system is assumed to be a DC motor to allow for a linear motor model. It is known that the motor follows the following formulas in Laplace domain:

$$T(s) = i(s)k_t \tag{6.2}$$

$$\omega(s) = \frac{T(s) - \omega(s)F}{Is} \tag{6.3}$$

$$V(s) = (R + Ls)i(s) + \epsilon(s) \tag{6.4}$$

$$\epsilon(s) = \omega(s)k_e \tag{6.5}$$

$$k_e = k_t \tag{6.6}$$

Here $T(s)$ is the torque delivered by the motor, $i(s)$ is the DC current, k_t is the torque constant, I is the rotational inertia, L is the terminal inductance, R is the terminal resistance, $\epsilon(s)$ is the back-emf, and k_e is the back-emf-constant. Thus:

$$\frac{\omega(s)}{V(s)} = \frac{k_t}{ILs^2 + (RI + LT_f)s + k_e k_t} \tag{6.7}$$

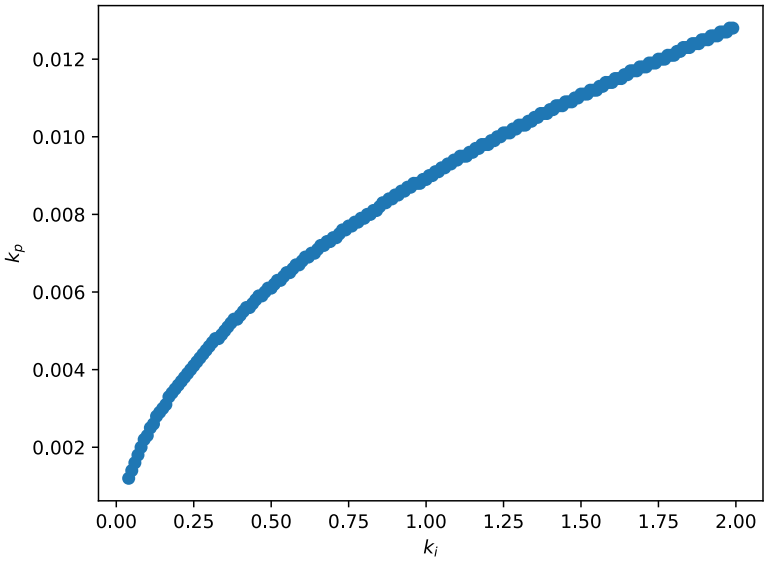

Figure 6.2: The values for k_p for which the system is critically dampened.

And the transfer function of the PI controller can be written as follows:

$$C(s) = k_p + \frac{k_i}{s} = \frac{k_p(s-c)}{s} \tag{6.8}$$

With $c = -\frac{k_i}{k_p}$. This makes c the position of the zero. This makes the transfer function of the entire system equal to:

$$G(s) = C(s) \cdot 50 \cdot \frac{\omega(s)}{V(s)} \tag{6.9}$$

The 50 term is added to denormalize the voltage, as the PI controller implemented must output normalized voltage. A script is created to determine the exact value of k_p for which the system is critically dampened for each value of k_i. Which creates the plot shown in Figure 6.2. The higher the value chosen for k_i, the faster the system responds. Thus, it makes sense to choose a large value for k_i, however, when this value is too large, two issues occur: Due to oscillations from the velocity estimator, oscillations also are transferred to the output power. This phenomenon scales with k_p. The integral factor of the PI controller is bounded to prevent it from growing to infinity for unreachable velocities, thus, the ideal PI controller used in the transfer function becomes less accurate for large values for k_i. Testing has shown that $k_i = 0.3$ creates a plot with limited overshoot, while it still reacts quickly to changes. This corresponds with $k_p = 0.0046$. This system also is not too effected by the issue with the oscillating velocity measurements. The resulting velocity plot is shown in Figure 6.3.

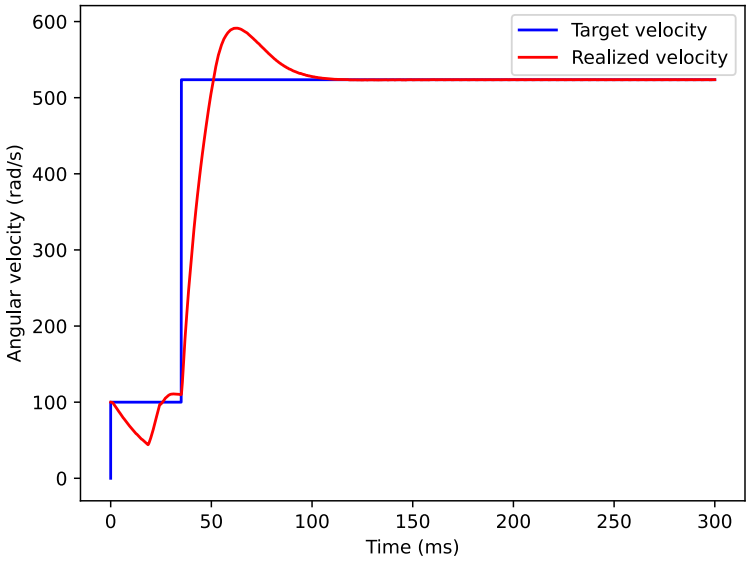

Figure 6.3: Simulations using the tuned PI controller.

6.1.4 Current limiting

The electronics used and the motor have a maximum current rating. To make sure this is followed, a current limiter is implemented. This is implemented by making use of the current sensors. The following equation is used to calculate the total current flow:

$$i_{total} = \frac{|i_a| + |i_b| + |i_c|}{2} \tag{6.10}$$

If the total current is larger than 180 A, the controller sets two of the terminals to floating to cut the power. A maximum of 150 A has been chosen to allow for a 30 A margin. A margin is required as the current may increase between update cycles of the controller, and as the current measurements may have an error.

6.2 Field oriented control

This section explains how the PI controllers can be tuned in such a way that they do not interfere, how the observer can be implemented, and why a certain type of vector space modulation should be implemented.

6.2.1 PWM modulation scheme

Figure 6.4 shows where PWM modulation takes place in the FOC controller. As mentioned in chapter 5, the modulation scheme that was used for the first simulations is sinusoidal PWM (SPWM).

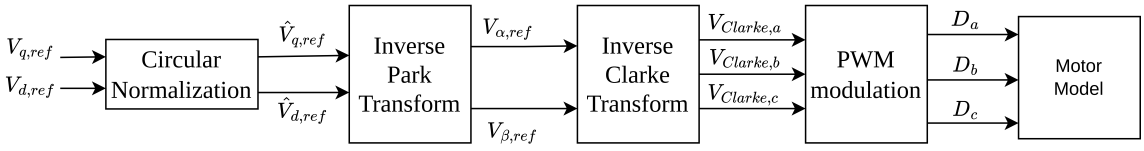

Figure 6.4: PWM modulation in the FOC controller

For SPWM, the PWM modulation block in Figure 6.4 consists of a simple scaling. SPWM, however, does not make optimal use of the DC voltage bus (V_{DC}). The outputs of the Clarke transform are sinusoidal waveforms varying from -1 to 1. Since duty cycles cannot be negative, these waveforms are scaled to sinusoids varying from 0 to 1, where 1 corresponds to a duty cycle of 100% and thus a terminal voltage equal to V_{DC}. Equation 6.11 shows how the phase voltages can be computed from the terminal voltages.

$$V_{ph,x} = V_{t,x} - V_c \tag{6.11}$$

Here, $V_{ph,x}$ is the phase voltage of the A, B or C phase in V, $V_{t,x}$ is the terminal voltage of the A, B or C terminal in V and V_c is center voltage in V. V_c is calculated as follows:

$$V_c = \frac{V_{t,a} + V_{t,b} + V_{t,c}}{3} \tag{6.12}$$

Where $V_{t,abc}$ are the terminal voltages of the motor A, B and C terminal in V. Figure 6.5 shows the terminal voltages and one of the phase voltages resulting from SPWM. For the sake of legibility, only the phase A voltage is shown. The other phase voltages are exactly the same waveform, but with a phase offset.

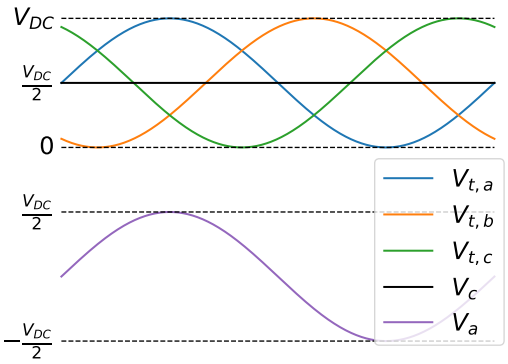

Figure 6.5: Influence of SPWM on the terminal and phase voltages

Since V_c is the average of the three terminal voltages, it can be seen as a DC voltage equal to $\frac{V_{DC}}{2}$. It is easy to see how subtracting this DC offset from the terminal A voltage, as per Equation 6.11, will lead to the sinusoidal phase voltage as shown in Figure 6.5.

Space Vector Pulse Width Modulation (SVPWM)

By using an improved PWM technique, the amplitude of the phase voltage can be increased. The modulation scheme in question is called SVPWM [23]. Figure 6.6 shows a schematic of how this modulation scheme is applied.

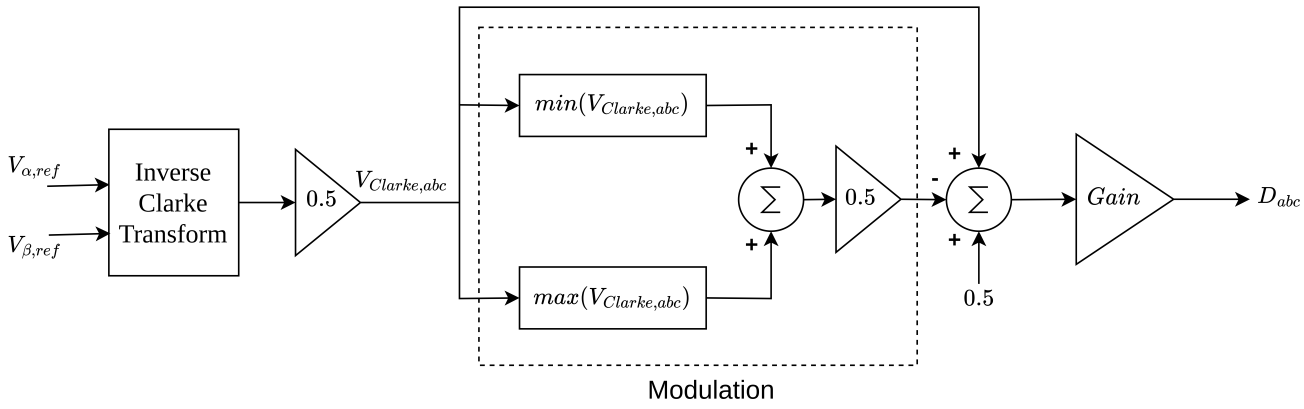

Figure 6.6: Schematic representation of SVPWM

If the modulation and the gain block were omitted, the terminal voltages would have looked as shown in Figure 6.7. Note that this would have been equivalent to applying the SPWM technique. The average of the minumum and maximum terminal voltage is shown by the black triangular waveform in Figure 6.7. This waveform is equivalent to the output of the modulation module in Figure 6.6 if it were scaled to V_{DC}.

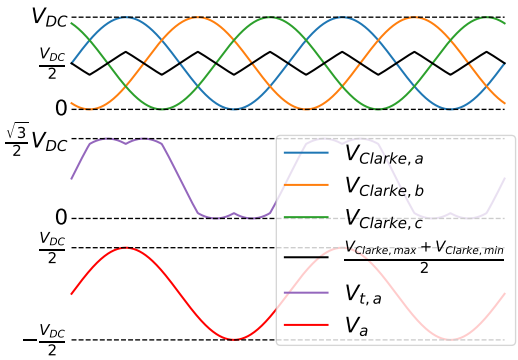

Figure 6.7: Influence of SVPWM on the terminal and phase voltages

When this average is subtracted from the outputs of the scaling module, a waveform such as the purple waveform shown in Figure 6.7 is obtained. If this output would have been directly fed to the motor, the terminal voltages would never rise above $\frac{\sqrt{3}}{2}V_{DC}$. However, this would still result in phase voltages varying from $-\frac{V_{DC}}{2}$ to $-\frac{V_{DC}}{2}$, as shown by the red waveform in Figure 6.7. Since the terminal voltages can use the whole V_{DC} range, it is possible to add a gain factor as shown in Figure 6.6. This gain factor is equal to $\frac{2}{\sqrt{3}}$, and allows for the terminal voltages to rise to V_{DC}. Looking at Equation 6.11 and Equation 6.12, scaling the terminal voltages up by a factor $\frac{2}{\sqrt{3}}$, results into the terminal voltages being scaled up by the same factor. Thus, the SVPWM allows the terminal voltages to reach to $\frac{V_{DC}}{\sqrt{3}}$, which is an increase in voltage range of approximately 15% with respect to SPWM. Therefore, SVPWM makes better use of the DC bus voltage.

Improved SVPWM

An improved version of SVPWM was proposed in the reference code [12]. The schematic of this modulation scheme is shown in Figure 6.8.

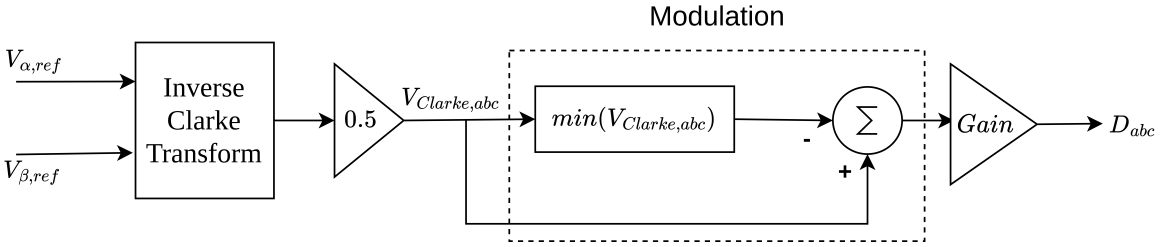

Figure 6.8: Schematic representation of the improved version of SVPWM

It can be seen that this schematic is very similar to the schematic for SVPWM shown in Figure 6.6, but contains a few simplifications. Instead of subtracting the average of the maximum and minimum values of $V_{Clarke,abc}$ from $V_{Clarke,abc}$, the minimum of $V_{Clarke,abc}$ is subtracted. The minimum is calculated in the modulation module and is shown as the black waveform in Figure 6.9. The signals are scaled to from Figure 6.9 are scaled to V_{DC} in the same way as in Figure 6.7.

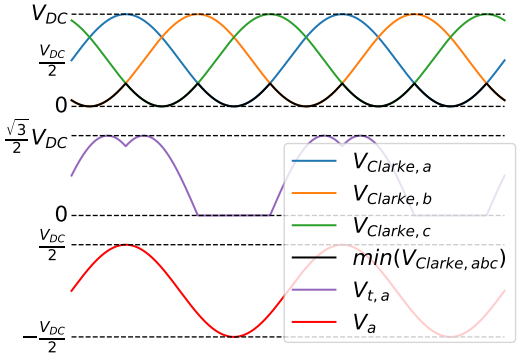

Figure 6.9: Influence of improved SVPWM on the terminal and phase voltages

It can be seen that this simplified modulation scheme yields the same results as SVPWM. The terminal voltage in Figure 6.9 has the same magnitude and waveform as in Figure 6.7, thus, following the same logic as for SVPWM, the gain factor can be set to $\frac{2}{\sqrt{3}}$. This will result into an increase in voltage range of approximately 15% for the phase voltages. The advantage of using the simplified SVPWM modulation scheme becomes apparent when the terminal voltages as shown in Figure 6.7 and Figure 6.9 are compared. It can be seen that the voltage of the motor A terminal in Figure 6.9 stays at zero for a bit lass than half of the period of the A phase voltage. This means that that in the A leg of the inverter circuit, the lower switch remains closed during this time. Thus, in comparison to SVPWM, this improved SVPWM method reduces the switching losses. This is the reason why it was chosen to implement improved SVPWM for FOC.

6.2.2 Observer in $\alpha\beta$ reference frame

An observer needs to be implemented in order to estimate the electrical angle of the rotor. For the implementation of FOC, two observers were considered: the Leuenberger observer [24] and the Sliding Mode Observer (SMO) [25]. The two observers have a comparable performance, but have their own drawbacks. The SMO suffers from chattering due to its nonlinear character. This means that the signal for the angle estimate will contain high frequency oscillations. Since the current controller is band-limited, it is not able to react fast enough on these changes. This may result into oscillatory behaviour of the current. The Leuenberger observer is linear, which means that it does not suffer from chattering, and is easier to tune. However, it has a less good parameter robustness [26]. In order to tune the Leuenberger observer, it is necessary to know the phase resistance and phase inductance of the motor. The phase resistance, however, changes with the temperature. Thus, in order to implement the Leuenberger observer it would have been necessary to model the temperature changes in the motor together with according change in the phase resistance. This was considered out of scope for this thesis, and thus it was decided to implement the SMO.

In order to implement the SMO, the motor phase circuits in the two-phase ($\alpha\beta$) reference frame were considered. Since the two phase circuits are the same, they are shown in one figure:

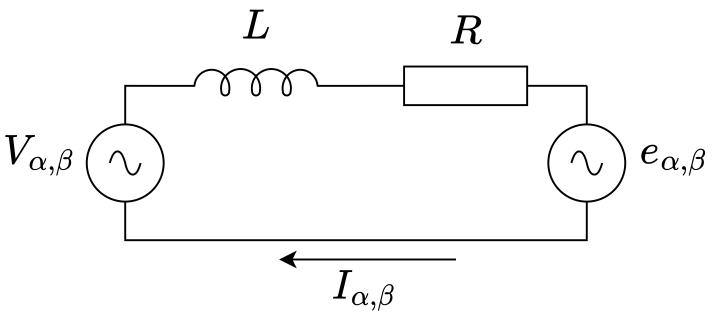

Figure 6.10: Phase circuit in the $\alpha\beta$ reference frame. R is the phase resistance in Ω, L is the phase inductance in H, $e_{\alpha,\beta}$ is the α or β back electromotive force (back-emf) in V, $I_{\alpha,\beta}$ is the α or β phase current in A and $V_{\alpha,\beta}$ is the respective α or β phase voltage in V. Note that $V_{\alpha,\beta}$ and $e_{\alpha,\beta}$ are always in phase.

The relevant equations for the circuits shown in Figure 6.10 are as follows:

$$\begin{bmatrix} v_\alpha(t) \\ v_\beta(t) \end{bmatrix} = R \cdot \begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix} + L\frac{d}{dt}\begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix} + K_e\omega(t)\begin{bmatrix} -sin(\theta(t)) \\ cos(\theta(t)) \end{bmatrix} \tag{6.13}$$

Here, R is the phase resistance in Ω, L is the phase inductance in H, $I_{\alpha,\beta}$ are the phase currents in A and $V_{\alpha,\beta}$ are the phase voltages in V. ω is the electrical angular velocity of the rotor in $\mathrm{rad\,s^{-1}}$, K_e is the back-emf constant and θ is the electrical angle in rad. The last term in the equation are the back electromotive forces (back-emf). It can be seen that the back-emf is dependent on the rotor angle. Thus, to extract the rotor angle, it is first necessary to estimate the back-emf. This is done through estimation of the phase currents. First, the phase current is estimated with some assumed value for the back-emf. Then, the estimated phase current is compared to the phase current that has been measured by the sensors in the inverter circuit and transformed to the α, β reference frame. Finally, the error signal is calculated and fed to a controller, which tries to make the error signal converge to zero by adjusting the assumed back-emf value. In the case of the SMO, this is a sliding-mode controller. In order to estimate the phase currents, Equation 6.13 is first transformed to the Laplace domain. Since the equations for the α and β circuits are equivalent, the subscripts are dropped:

$$I(s) = \frac{V(s) - e(s)}{R + sL} \rightarrow I(s) = (V(s) - e(s)) \cdot \frac{1}{R} \cdot \frac{1}{1 + \frac{L}{R}s} \tag{6.14}$$

It can be seen that the phase current can be modelled by passing the signal $(V(s) - e(s))$ through a first-order lowpass filter. This filter has been modelled as a digital Butterworth filter with a cutoff frequency of $\frac{R}{2\pi L}$ Hz and a sample frequency of 100 kHz, which is equal to the switching frequency. The SMO model is shown in Figure 6.11.

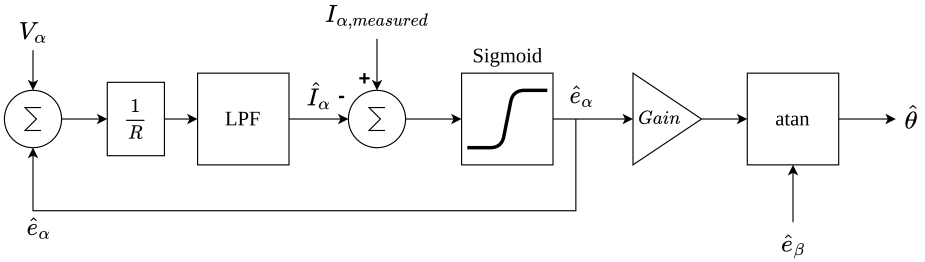

Figure 6.11: Scheme of the SMO for the α phase. The emf of the beta phase is estimated with the same observer structure and therefore not shown in the figure.

The classical SMO utilizes a signum function [27]. For the classical observer, a lowpass filter needs to be applied after the gain factor to reduce some of the chattering. However, this would cause a delay and result in a phase shift [28]. Therefore, instead of implementing the sign function, it has been decided to implement a sigmoid. The function of the sigmoid is given as:

$$sigm(\epsilon) = \frac{2}{1 + e^{-\alpha\epsilon}} - 1 \tag{6.15}$$

Here, ϵ is the error and α is a factor that specifies how smooth the function is. The larger α the more the sigmoid function approximates the signum function. The gain factor as well as the α factor have to be tuned in order to make the SMO function correctly. Since the mathematical analysis of a non-linear control algorithm was considered to be out of scope for this thesis, the SMO has been tuned experimentally. However, it has been attempted to give an intuitive explanation on how the parameters influence the behaviour of the SMO. The α parameter, in a sense, sets the threshold of the error that the SMO will react to. The smoother the sigmoid function, the larger the error has to be before the observer applies corrective action. If the sigmoid is smooth, the error will converge to an offset error, instead of zero. A benefit of smoothening the sigmoid is that this reduces the chattering. The gain of the observer determines how aggressively the corrective action is applied. If the observer reacts too aggressively, the error will oscillate. However, if the gain is too small, the error might not converge. In order for the SMO to be stable, the gain parameter has to adhere to the following criterion [25]:

$$Gain > max(|e_\alpha|, |e_\beta|) \tag{6.16}$$

Here, $e_{\alpha,\beta}$ are the α, β back-emf voltages in V. Now that the back-emf signals in the α, β-frame are estimated, it is possible to calculate the rotor angle as per Equation 6.17.

$$\theta = -arctan(\frac{e_\alpha}{e_\beta}) \tag{6.17}$$

In order to assess how well the SMO is able to estimate the rotor angle, the error between the real angle and estimated angle has been plotted in Figure 6.12. As can be seen, the SMO has an offset error. Although this is suboptimal, it has been decided to keep the observer parameters as is. This is due to the fact that removing the offset will come at the cost of increased chattering.

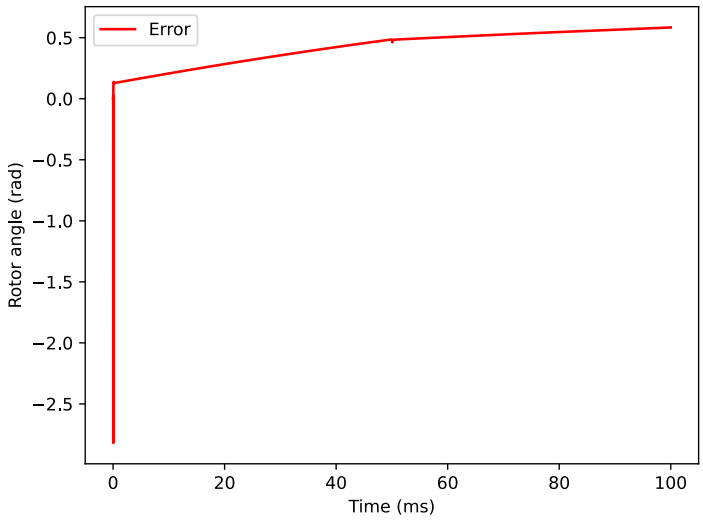

Figure 6.12

6.2.3 Designing the current PI controller

In order for FOC to be able to follow a reference current, a current controller needs to be designed. The standard implementation of the current controller is by means of a PI controller. The differential gain has been omitted due to its sensitivity to noise. Since FOC is controlling two currents (the direct and quadrature current), two PI controllers need to be designed. In this section, only the quadrature PI controller will be discussed, as the direct PI controller is completely equivalent. The controller has been tuned according to the method in [29] and [30].

Setting the integral and proportional gains

In order to tune the PI controller, a series model as shown in Figure 6.13 was considered.

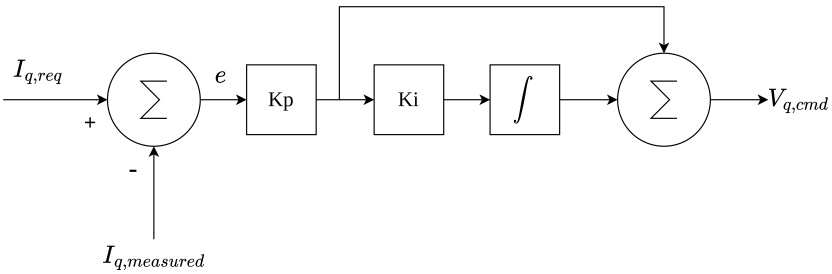

Figure 6.13: Series configuration of the current PI controller

In a series configuration, the proportional (K_p) and integral (K_i) gains correspond to tangible system parameters: K_p sets the gain and K_i determines the inflection point of the controller in rad/s. This can be seen in the transfer of the PI controller:

$$PI(s) = \frac{K_p K_i (1 + \frac{s}{K_i})}{s}) \tag{6.18}$$

Figure 6.14 shows a model of the small-signal model of the motor quadrature winding.

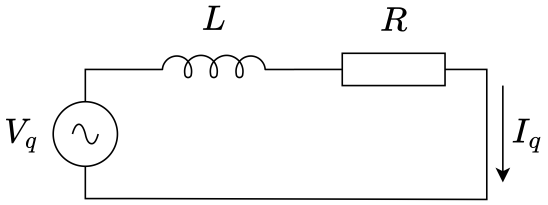

Figure 6.14: Small-signal model of the quadrature winding of the motor. V_q is the quadrature voltage in V, L is the quadrature inductance and R is the quadrature resistance. Since the back-emf changes slowly with the current, it is assumed to be constant and thus ignored for the small-signal model.

In this system, the voltage is the commanded duty cycle that was scaled to the DC bus voltage (V_{DC}) and the current is what needs to be controlled. The transfer function for this system is represented in Equation 6.19.

$$\frac{I_q(s)}{V_q(s)} = F_s \cdot \frac{\frac{1}{R}}{1 + \frac{L}{R}s} \tag{6.19}$$

$$F_s = G\frac{V_{DC}}{2} \tag{6.20}$$

F_s is the scaling factor and G is the gain. The gain depends on the PWM modulation scheme that is being used, as exlained in subsection 6.2.1. For Sinusoidal PWM, for example, G would be equal to 1. This is due to the fact that the output duty cycles vary from -1 to 1 and result in phase voltages varying from $-\frac{V_{DC}}{2}$ to $\frac{V_{DC}}{2}$. Since for SVPWM the phase votages can reach to $\frac{V_{DC}}{\sqrt{3}}$, the gain G for this modulation technique will be equal to $\frac{2}{\sqrt{3}}$.
The open loop and closed loop gain are defined as follows:

$$G_{open-loop} = PI(s) \cdot \frac{I_q(s)}{V_q(s)} \tag{6.21}$$

$$G_{closed-loop} = \frac{G_{open-loop}}{1 + G_{open-loop}} \tag{6.22}$$

By substituting Equation 6.21 Equation 6.18 in Equation 6.22, and doing some algebraic manipulation, Equation 6.23 is obtained.

$$G_{closed-loop} = \frac{1 + \frac{1}{K_i}s}{\frac{L}{F_s K_i K_p}s^2 + (\frac{1}{K_i} + \frac{R}{F_s K_i K_p})s + 1} \tag{6.23}$$

It is desirable to select the gains of the PI controller in such a way that all poles are real, since complex poles can introduce resonance in the control system. For the poles to be real, it should hold that the denominator of the closed loop gain can be written as described by Equation 6.24.

$$\frac{L}{F_s K_i K_p}s^2 + (\frac{1}{K_i} + \frac{R}{F_s K_i K_p})s + 1 = (1 + Cs)(1 + Ds) \tag{6.24}$$

By working this out, Equation 6.25 and Equation 6.26 are obtained.

$$CD = \frac{L}{F_s K_p K_i} \tag{6.25}$$

$$C + D = \frac{1}{K_i} + \frac{R}{F_s K_i K_p} \tag{6.26}$$

By substituting C and D as per Equation 6.27 and Equation 6.28, and rewriting Equation 6.23, the closed loop gain can be written in the form of Equation 6.29.

$$C = \frac{R}{F_s K_i K_p} \tag{6.27}$$

$$D = \frac{1}{K_i} \tag{6.28}$$

$$G_{closed-loop} = \frac{1 + \frac{s}{K_i}}{(1 + \frac{R}{F_s K_i K_p}s)(1 + \frac{s}{K_i})} = \frac{1}{1 + \frac{R}{F_s K_i K_p}s} \tag{6.29}$$

It appears that by choosing C and D correctly, it is possible to not only end up with real poles, but to also make a pole cancel out a zero. By solving Equation 6.25 with the aforementioned substitutions for C and D, Equation 6.30 is obtained.

$$K_i = \frac{R}{L} \tag{6.30}$$

By substituting this value for K_i in Equation 6.29, Equation 6.31 is obtained. Now, it can be seen that the bandwidth of the system is set by K_i.

$$G_{open-loop} = \frac{1}{1 + \frac{L}{F_s K_p} s} \tag{6.31}$$

Until now, the discontinuity of the system has not been taken into account. Since the quadrature voltage is commanded during one PWM cycle, but only has effect in the next PWM cycle, a delay is present. From a system modeling perspective, this looks like a sample-and-hold function with a sampling frequency equal to the PWM update rate frequency [30]. This can be modeled as a delay of half of our PWM cycle, as shown in Equation 6.32, where T_PWM is the PWM period in seconds.

$$G_{delayed} = G_{closed-loop} \cdot e^{-\frac{T_{PWM}}{2} s} \tag{6.32}$$

For a closed-loop gain of unity for all frequencies and a PWM period unity, the phase lag would look like it is shown in Figure 6.15. It can be seen that the phase lag gets exponentially worse for higher frequencies.

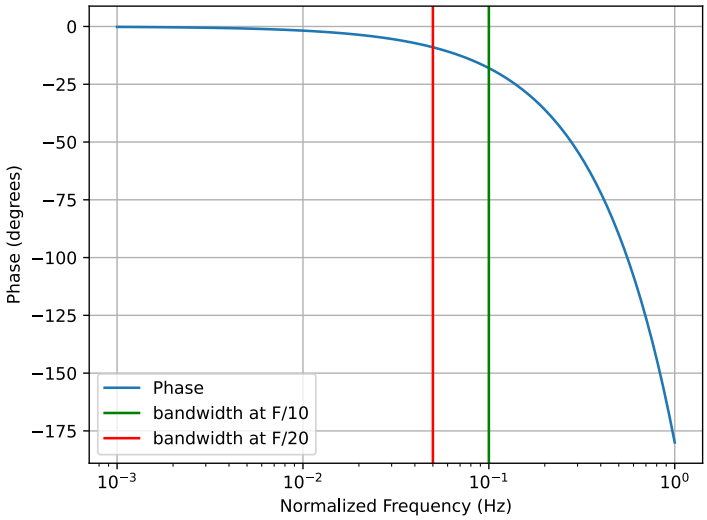

Figure 6.15: Phase lag caused by the sample-and-hold delay

it is desirable to limit the bandwidth of the closed-loop gain in such a way, that the system is not affected by the phase delay. However, the bandwidth should not be too low either, since that would make the PI controller too slow. Therefore, the bandwidth should be limited at a maximum $\frac{1}{10}$ of the switching frequency. This is represented by the green line in Figure 6.15 and corresponds to a maximum phase lag of about 18 degrees. however, it was chosen to take some more margin, and to set the bandwidth at $\frac{1}{20}$ of the switching frequency. This is shown with the red line in Figure 6.15 and corresponds to a maximum phase lag of 9 degrees. By imposing the condition that the pole of the closed-loop gain should occur at a frequency equal to $\frac{1}{20}$ of the switching frequency, it is possible to determine K_p as per Equation 6.33.

$$\frac{F_s K_p}{L} = \frac{f_{PWM}}{20} \Rightarrow K_p = \frac{L \cdot f_{PWM}}{F_s \cdot 20} \tag{6.33}$$

PI controller saturation

If the commanded quadrature voltage is within the bounds of $-G\frac{V_{DC}}{2}$ to $-G\frac{V_{DC}}{2}$, the PI controller will work well. However, it could happen that a quadrature current is requested, which can only be generated with a quadrature voltage higher than $-G\frac{V_{DC}}{2}$. In such a case, the output of the PI controller will clamp at 1 and its operation will not be linear anymore. However, the integral term will continue to grow, since the error signal cannot be reduced to zero. When the controller exits saturation mode, the large integral factor can cause an overshoot that will

destabilize the PI controller. Thus, the growth of the integral factor must be prevented by setting limits. Since the outputs of the controller varies from -1 to 1, it was chosen to set the limits for the integral factor at -1 and 1.

Results

From Figure 6.16, it can be seen that the PI current controllers are able to follow the requested quadrature and direct currents. For the first 50 ms, FOC was commanded to reach the maximum torque, which corresponds to a quadrature current of about -60 A. Please note that the minus sign is present due to the axis and positive rotation definition as shown in Figure 5.8. After approximately 60 ms, reaching the requested current becomes impossible, due to the increase of the back-emf. After the target current is set to a value which can be reached, it can be seen that the current controller are able to restore and follow. This means that limiting the I term as was proposed in section 6.2.3 has the desired effect.

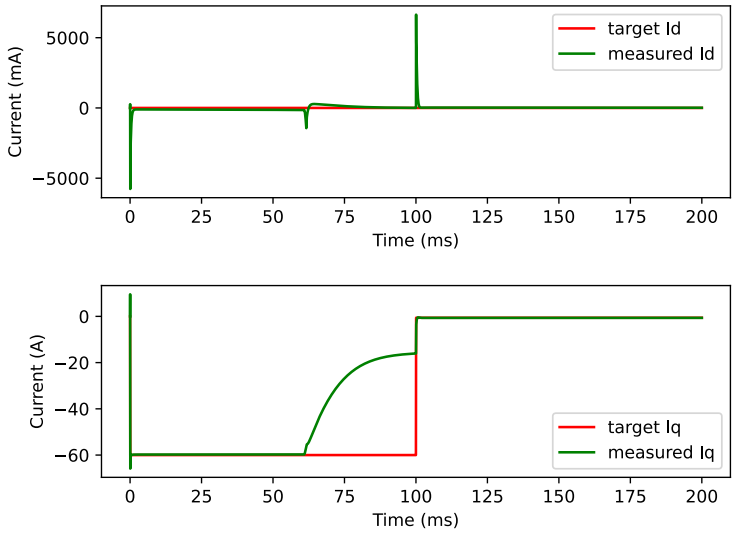

Figure 6.16: The PI controllers are able to follow the requested currents.

6.2.4 Velocity controller

As there is no measurement for velocity, a controller must be chosen that does not require this. It may seem like another solution would be to take the derivative of the position estimate, however, this is not accurate due to chattering. A way to solve this is by making use of a phase-locked loop (PLL). Although an implementation of a PLL has been attempted Appendix C, it does not function due to unknown bugs. Due to time restrictions, this issue has not yet been solved.
To design a controller that does not require a phase-locked loop, the equation for rotational velocity is required:

$$\omega(t) = \int_0^t \frac{T(t) - \omega(\tau) \cdot F}{I} \, d\tau \tag{6.34}$$

Here $\omega(t)$ is the rotational velocity, t is time, $T(t)$ is torque, F is the coefficient of friction, and I is the rotational inertia.
Thus, if steady-state is assumed, $T - \omega F = 0$. Thus, $T = \omega F$. If taking $T = \omega_{desired} F$ results in Equation 6.34 converging to $\omega_{desired}$, this controller is stable.

6.3 Benchmarking

In order to compare the two control algorithms that were proposed in section 6.1 and section 6.2, benchmarks were designed. These benchmarks are aimed at testing whether the control algorithm in question adheres to the requirements, as mentioned in chapter 2. The simulation results for each benchmark can be seen in Appendix I.

6.3.1 Maximum velocity with a small propeller

To test if the required velocity can be reached, two simulations are performed. For one of the simulations, the desired velocity is set to the velocity that must be reached. For the other benchmark, the desired velocity is set to double the velocity that must be reached.

Trapezoidal control is not able to converge to the desired velocity for the simulation with the lower desired speed. It falls slightly short, the reason for this is explained in subsection 6.1.3. However, the velocity requirement is met for the simulation with the desired velocity set higher.
The results are very similar for FOC. It gets close to reaching the required velocity for the simulation with a lower desired velocity, but it is able to just hit the requirement for the other simulation. Theoretically, FOC is supposed to reach an even higher maximum rotational velocity as the maximum voltage range is higher. It was also noticed that the velocity keeps oscillating instead of converging. This is due to how the observer is tuned, as the oscillating error translates to oscillations in velocity.

6.3.2 Various starting velocities

This benchmark was designed to test if the control algorithm is able to reach a desired velocity from different starting velocities. For this test, the target velocity was set to $600\,\mathrm{rad\,s^{-1}}$. In total, three simulations were run: for the first one, the starting velocity was set to zero, for the second one it was set to $300\,\mathrm{rad\,s^{-1}}$ and for the third one to $900\,\mathrm{rad\,s^{-1}}$. Although not every possible starting velocity in the motor operating range has been tested, these simulations will give an indication of whether the control algorithm adheres to the requirement under test.
Trapezoidal control is able to converge to the target velocity without issue, however, it does have an overshoot as is explained in subsection 6.1.3. The simulation results seem to indicate that trapezoidal control adheres to the requirement.
FOC converges with an offset and thus does not reach the target speed. The rate of convergence is also much slower than is the case with trapezoidal control. For a zero starting speed, it takes a lot of time for FOC to start, due to the absence of a starting sequence and thus reliance on the observer while the back-emf signal is very small. It is expected that in reality FOC will not be able to start from zero at all, due to the presence of measurement noise, which has not been simulated. Thus, FOC does not adhere to the requirement.

6.3.3 Faulty measurements

It is difficult to determine if the control algorithms can handle every faulty measurement, as edge cases can always occur. To get an indication if this requirement is met, a simulation was performed where the controller runs at $600\,\mathrm{rad\,s^{-1}}$. Then, for 200 ms all measurements are returned as zero, whereafter they are restored.
For trapezoidal control, the velocity falls to zero and never recovers. It, thus, fails this test.
FOC fairs a lot better. Although the velocity does fall rapidly in the period without measurements, it starts to recover after the recovery measurements.

6.3.4 Sudden change in velocity

To determine if the controller is able to handle a sudden change in velocity, a test is performed where the controller runs at $600\,\mathrm{rad\,s^{-1}}$. Then, the velocity is suddenly reduced by $75\,\mathrm{rad\,s^{-1}}$.
Both trapezoidal control and FOC are able to handle this change without issue. After the reduction in velocity, both of the controllers speed back up to the desired velocity. In the case of FOC, it converges to back the desired velocity with an offset.

6.3.5 Efficiency in steady state

In order to give an indication of the efficiency of the control algorithms, the motor was brought to a steady state. This means that the motor was spinning at a set target velocity for a prolonged period of time. When the velocity converged to the target velocity, the efficiency of the control algorithm was calculated.
For trapezoidal control, the efficiency was calculated to be 98%. For FOC, this was calculated to be 97%. However, due to the fact that FOC converges to a velocity that is higher than the target velocity, the efficiency was in fact calculated for a slowly accelerating motor. It can be concluded that both FOC and trapezoidal control adhere to the requirement of having a 70% efficiency.

6.3.6 Output realized angular velocity

In each of the benchmarks that were run, other than the one for faulty measurements, it is shown that trapezoidal control outputs the realized rotational velocity at all times. The proposed setup for FOC is not able to do this.

6.3.7 Limiting gate currents

For both FOC and trapezoidal control, the gate currents never exceed 180 A in any of the benchmarks. If it would ever be exceeded, it would be expected that this would occur at either the velocity change being maximum, thus, the first benchmark, or when faulty measurements occur. Since trapezoidal control contains a current limiter, it can be concluded that it adheres to this requirement. For FOC, more tests need to be conducted, such as

requesting an extremely high torque for the controller. It cannot be concluded yet whether or not FOC adheres to this requirement.

6.3.8 Inherently tested requirements

As the simulator inherently test requirements 7, 8, 9, and 11 from chapter 2 due to the way it is built, it can be concluded that these requirements are adhered to by both the implementations of trapezoidal control and FOC.

Chapter 7

Conclusion and Outlook

This chapter explains which algorithm was chosen and why. It gives recommendations on changes and improvements can be made on the chosen algorithm. And what further research can be conducted.

7.1 Chosen algorithm

As became apparent in section 6.3, FOC as implemented does not adhere to some of the most essential requirements from chapter 2, such as being able to start up from standstill. Therefore, the proposed design of the trapezoidal control algorithm will be implemented on the hardware created by the other subgroups.
As discussed in section 6.3, this design adheres to the requirements 3, 4, 5b, 13 from the chapter 2. Since requirements 7, 8, 9, and 11 are inherent parts of the simulator, it follows that trapezoidal control adheres to these requirements too. The simulations also show that requirements 10 and 12 are adhered to. However, the design fails to meet requirements 5a, 6, and 14, and requirement 5c has yet to be been tested. Thus, the current design requires some improvements.

7.2 Improvements and alternatives

7.2.1 Accurate velocity estimation

One of the most prominent disadvantages of the current implementation of trapezoidal control is that, for higher target velocities, the controller has an offset. This is most likely due to inaccuracies in the velocity estimation due to limited resolution in the way it is calculated. It would be recommended to find a different method to determine the velocity if one were to use this system. This would also allow for retuning the PI controller to have less overshoot, as the PI controller in this report considered the extra offset caused by increasing the P factor due to this issue.

7.2.2 FOC

As was concluded in section 6.3, the proposed implementation for FOC does not adhere to requirements 4 and 10 as stated in chapter 2. Requirement 12 has not been tested. Therefore, in the current state, it is unfit to be implemented on the hardware. The main advantage that FOC has over trapezoidal control is that it can in theory reach higher angular velocities. As trapezoidal control is able to reach the required velocity as stated in chapter 2, this advantage is not relevant for this thesis. However, if the reader is interested in elevating the speed limit of the motor, they might be interested in improving the FOC algorithm as proposed in section 6.2 instead of using the proposed trapezoidal controller.

Start-up sequence

A start-up sequence should be implemented. The start-up can be realized by implementing a V/f controller or an I-f controller. The I-f start-up method uses the current controller as implemented in FOC to accelerate the motor at a constant torque. Because I-f and FOC are using the same controller, the transition to sensorless FOC from the start-up sequence is simpler [31].

Estimation of the angular velocity

As mentioned in section 6.2, the FOC algorithm as currently implemented is not able to estimate the angular velocity of the rotor. The estimation of the angular velocity can be realized by implementing a phase-locked loop

(PLL) which, as input, makes use of the estimated angle from the sliding mode observer. Although an attempt at implementing the PLL has been made Appendix C, it does not yet function and needs debugging.

Velocity offset

As was seen in section 6.3, the realized velocity of the current FOC implementation has an offset with respect to the target velocity. This is due to the fact that currently, the target velocity is translated to a target torque, which is linearly dependent on the target quadrature current. However, as became apparent in subsection 6.2.2, the observer has an angle offset, which results into an error in the measured quadrature current. Therefore, the output torque of the motor will have an offset with respect to the requested torque, as the current controller inside the FOC algorithm assumes that the measured quadrature current is correct. Since the steady state velocity is linearly dependent on torque, a torque offset will translate to an offset in the steady state velocity. Therefore, the velocity offset as seen in section 6.3 is the cause of two underlying problems, namely: the lack of an adequate speed controller (which cannot be implemented due to the absence of an angular velocity estimate), and an error in the angle estimation.

Improving the angle estimation

The offset in the estimated angle is present due to the fact that the Sliding Mode Observer is not fast enough in reacting to the changes in the back-emf. This cannot be solved by making the sigmoid function as used in the SMO subsection 6.2.2 less smooth, since this would result into unacceptable oscillatory behaviour. It would, however, be possible to implement an observer in the rotor reference frame. In the dq reference frame, the oscillating back-emf signals will become DC signals that change more slowly, which means that the bandwidth of the observer can be reduced.

7.2.3 Modified trapezoidal modulation

As suggested in subsection 5.5.1, it could be worth investigating modified trapezoidal modulation further. Specifically, it could be worth investigating how this control algorithm could be made sensorless, as [10] explains that it has various benefits over standard trapezoidal control.

7.2.4 Variable loads

As drone manufacturers may be interested in using various propellers, it is worth finding a method to tune the controller to work well for a range of loads instead of a single load. For example, a system could be implemented to tune the controller while it is operating.

7.2.5 Better estimates for inertia and friction

For the simulations, the inertia, and friction were estimated using an approach that is not very accurate. To get more accurate results, and more accurate tuning parameters, these values can be determined in a more accurate manner.

7.2.6 Rotary encoder

Using a rotary encoder allows for a much simpler design of the controller, as no observer is required. It also is known to allow for more robust and efficient control. Thus, if budget allows, and if these factors are of importance, it is worth considering including it in the design of your hardware.

Bibliography

[1] T-MOTOR. "Mn501-s kv240." (2023), [Online]. Available: https://store.tmotor.com/goods-695-MN501-S+KV240.html.

[2] T-MOTOR. "P18*6.1 prop-2pcs/pair." (2023), [Online]. Available: https://store.tmotor.com/goods-384-P18%5C%2A61+Prop-2PCSPAIR.html.

[3] S. Dwivedi, M. Laursen, and S. Hansen, "Voltage vector based control for pmsm in industry applications," Jul. 2010. DOI: 10.1109/ISIE.2010.5637742.

[4] S. N. I. Boldea, *Vector Control of AC Drives*. CRC Press, Inc., 1992.

[5] A. M. Trzynadlowski, *Control of Induction Motors*. Academic Press, 1999.

[6] B. Akin and N. Garg, *Scalar (v/f) control of 3-phase induction motors*, Texas Instruments, 2013. [Online]. Available: https://www.ti.com/lit/pdf/sprabq8.

[7] B. Akin and M. Bhardwaj, *Sensorless field oriented control of 3-phase permanent magnet synchronous motors*, Texas Instruments, 2009. [Online]. Available: https://www.ti.Fcom/lit/an/sprabq3/sprabq3.pdf.

[8] C. Reza, M. D. Islam, and S. Mekhilef, "A review of reliable and energy efficient direct torque controlled induction motor drives," *Renewable and Sustainable Energy Reviews*, vol. 37, pp. 919–932, 2014, ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2014.05.067. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136403211400392X.

[9] B. Akin and M. Bhardwaj, *Sensorless trapezoidal control of bldc motors*, Texas Instruments, 2013. [Online]. Available: https://www.ti.com/lit/pdf/sprabq7a.

[10] K. Betto, S. Joryo, and T. Morizane, "Improvement of driving efficiency of pmsm by using modified trapezoidal modulating signal," in *2020 22nd European Conference on Power Electronics and Applications (EPE'20 ECCE Europe)*, 2020, P.1–P.10. DOI: 10.23919/EPE20ECCEEurope43536.2020.9215789.

[11] T-MOTOR. "P14*4.8 prop-2pcs/pair." (2023), [Online]. Available: https://store.tmotor.com/goods-380-P14%5C%2A48+Prop-2PCSPAIR.html.

[12] B. Akin and M. Bhardwaj. *foc_sim*. (2013). GitHub. Accessed: May 2, 2023. [Online]. Available: https://github.com/dbking77/foc_sim/commit/09e66057375f1e44aad173cf7ae0a5fb865ad28e.

[13] P. Perera, F. Blaabjerg, J. Pedersen, and P. Thogersen, "A sensorless, stable v/f control method for permanent-magnet synchronous motor drives," *IEEE Transactions on Industry Applications*, vol. 39, no. 3, pp. 783–791, 2003. DOI: 10.1109/TIA.2003.810624.

[14] M. Taoussi, "Modern improvement techniques of direct torque control for induction motor drives - a review," in *Protection and Control of Modern Power Systems*, vol. 4, 2019, P.11–P.23. DOI: 10.1186/s41601-019-0125-5.

[15] M. H. B. Mohamed KADJOUDJ Noureddine GOLEA, "Voltage switching tables for direct torque control of pmsm drive," in *Scientific Bulletin-University Politehnica of Bucharest*, vol. 69, 2007, P.24–P.36.

[16] T. Vyncke, R. Boel, and J. Melkebeek, "On extended kalman filters with augmented state vectors for the stator flux estimation in spmsms," Mar. 2010, pp. 1711–1718. DOI: 10.1109/APEC.2010.5433462.

[17] M. B. Bilal Akin and J. Warriner, *Trapezoidal control of bldc motors using hall effect sensors*, Texas Instruments, 2013. [Online]. Available: https://www.ti.com/lit/pdf/sprabz4.

[18] P. Damodharan and K. Vasudevan, "Sensorless brushless dc motor drive based on the zero-crossing detection of back electromotive force (emf) from the line voltage difference," *IEEE Transactions on Energy Conversion*, vol. 25, no. 3, pp. 661–668, 2010. DOI: 10.1109/TEC.2010.2041781.

[19] S. Sakunthala, R. Kiranmayi, and P. N. Mandadi, "A study on industrial motor drives: Comparison and applications of pmsm and bldc motor drives," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 537–540. DOI: 10.1109/ICECDS.2017.8390224.

[20] S. Derammelaere, M. Haemers, J. De Viaene, F. Verbelen, and K. Stockman, "A quantitative comparison between bldc, pmsm, brushed dc and stepping motor technologies," in *2016 19th International Conference on Electrical Machines and Systems (ICEMS)*, 2016, pp. 1–5.

[21] L. Parsa and H. Toliyat, "Five-phase permanent-magnet motor drives," *IEEE Transactions on Industry Applications*, vol. 41, no. 1, pp. 30–37, 2005. DOI: 10.1109/TIA.2004.841021.

[22] M. A. Abu Hassan, A. R. Abdullah, N. Bahari, and M. I. Mohd Sabri, "Efficiency comparison of trapezoidal and sinusoidal method for brushless dc motor drive," in *Recent Trends in Power Engineering*, ser. Applied Mechanics and Materials, vol. 785, Trans Tech Publications Ltd, Sep. 2015, pp. 248–252. DOI: 10.4028/www.scientific.net/AMM.785.248.

[23] R. Doncker, de, D. Pulle, and A. Veltman, *Advanced electrical drives : analysis, modeling, control* (Power systems), English. Germany: Springer, 2011, ISBN: 978-94-007-0179-3. DOI: 10.1007/978-94-007-0181-6.

[24] J. Brouwer, "Position sensorless control of the pmsm in a wide speed range," M.S. thesis, Delft University of Technology, Sep. 2020. [Online]. Available: http://resolver.tudelft.nl/uuid:d8078a05-bcfd-469e-9c71-a87b4b7edcc5.

[25] S. Masoumi Kazraji, R. Soflayi, and M. B. Bannae Sharifian, "Sliding-mode observer for speed and position sensorless control of linear-pmsm," *Electrical, Control and Communication Engineering*, vol. 5, May 2014. DOI: 10.2478/ecce-2014-0003.

[26] Y. Zhang, Z. Zhao, T. Lu, L. Yuan, W. Xu, and J. Zhu, "A comparative study of luenberger observer, sliding mode observer and extended kalman filter for sensorless vector control of induction motor drives," in *2009 IEEE Energy Conversion Congress and Exposition*, 2009, pp. 2466–2473. DOI: 10.1109/ECCE.2009.5316508.

[27] Z. Ma and X. Zhang, "Fpga implementation of sensorless sliding mode observer with a novel rotation direction detection for pmsm drives," *IEEE Access*, vol. 6, pp. 55 528–55 536, 2018. DOI: 10.1109/ACCESS.2018.2871730.

[28] Y. Zhao, W. Qiao, and L. Wu, "Compensation algorithms for sliding mode observers in sensorless control of ipmsms," in *2012 IEEE International Electric Vehicle Conference*, 2012, pp. 1–7. DOI: 10.1109/IEVC.2012.6183241.

[29] D. Wilson. "Teaching your pi controller to behave (part ii)." (), [Online]. Available: https://e2e.ti.com/blogs_/b/industrial_strength/posts/teaching-your-pi-controller-to-behave-part-ii.

[30] D. Wilson. "Teaching your pi controller to behave (part vi)." (), [Online]. Available: https://e2e.ti.com/blogs_/b/industrial_strength/posts/teaching-your-pi-controller-to-behave-part-vi.

[31] Z. Wang, K. Lu, and F. Blaabjerg, "A simple startup strategy based on current regulation for back-emf based sensorless control of pmsm," English, *I E E E Transactions on Power Electronics*, vol. 27, no. 8, pp. 3817–3825, 2012, ISSN: 0885-8993. DOI: 10.1109/TPEL.2012.2186464.

Appendix A

Determining the phase inductance of the motor

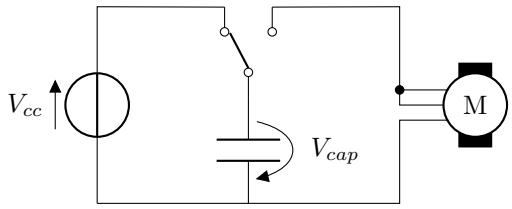

Figure A.1: The measurement setup used to determine the phase inductance of the motor. V_{cc} is set to 5 volts.

The value for L was determined using the measurement setup shown in Figure A.1. By closing the switch, an RLC circuit is created. The voltage over the capacitor is measured with an oscilloscope. Which should follow the equation given below:

$$V_{cap} = V_0 e^{-\alpha t} \frac{\alpha \sin(\omega t + \theta) + \omega \cos(\omega t + \theta)}{\alpha \sin(\theta) + \omega \cos(\theta)} \tag{A.1}$$

Here $t = 0$ is set to the time of switching. V_0 is the voltage of the capacitor at the time of switching. α is the attenuation ($\frac{R}{2L}$). ω is the angular resonance frequency ($\frac{1}{\sqrt{LC}}$). And, θ is the angle offset. This measurement was run twice with a 213 nF capacitor and three times with a 32.8 nF capacitor. Figures A.2, A.3, A.4, A.5, and A.6 show the resulting function fits of the values for V_0, R, L, and θ of Equation A.1 to the measurements. The average value for L is 16.93 µH. Thus, the phase inductance is 11.29 µH.

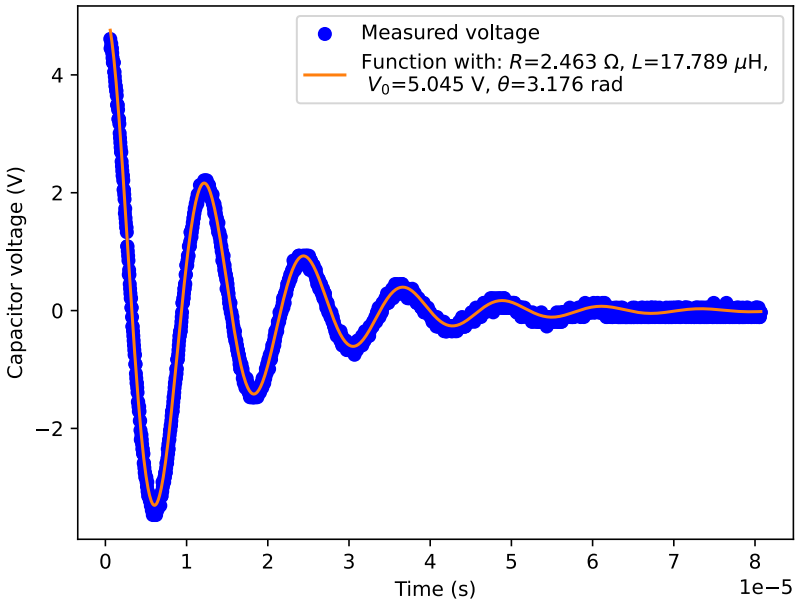

Figure A.2: Function fit with a 213 nF capacitor

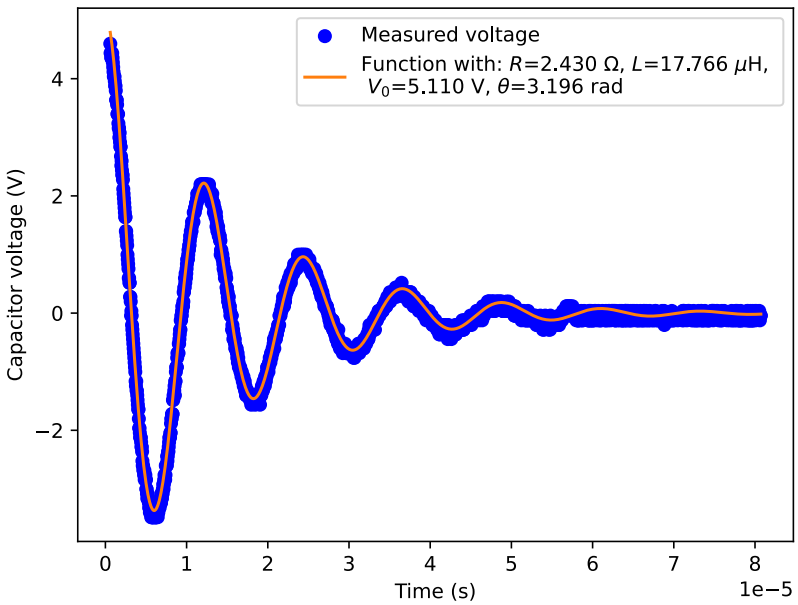

Figure A.3: Function fit with a 213 nF capacitor

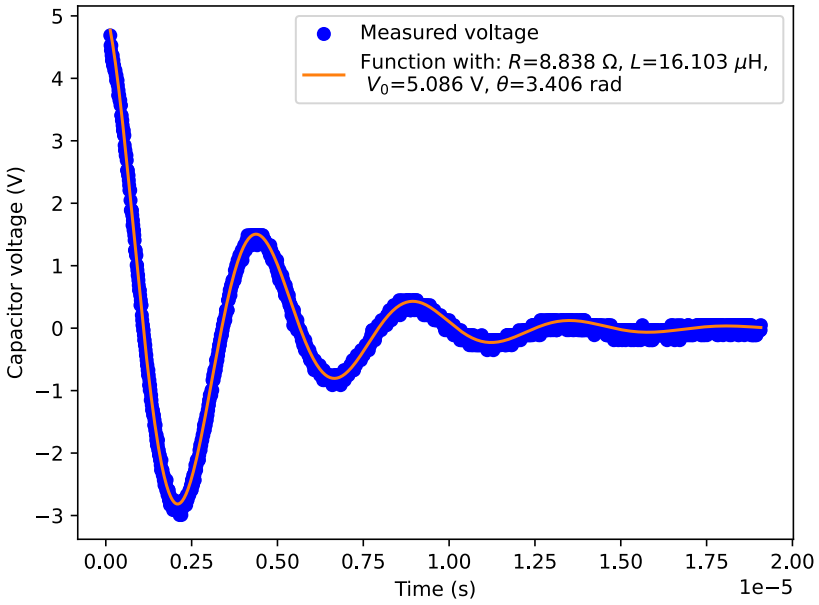

Figure A.4: Function fit with a 32.8 nF capacitor

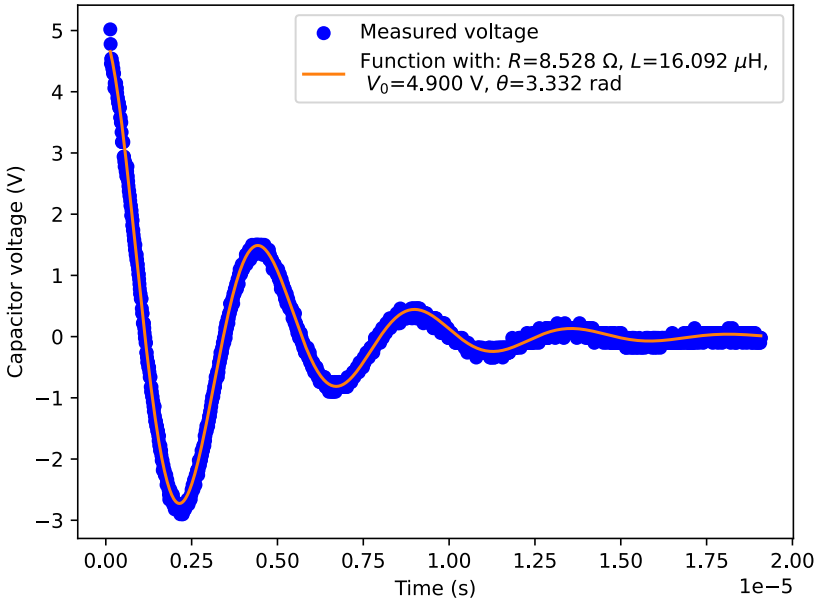

Figure A.5: Function fit with a 32.8 nF capacitor

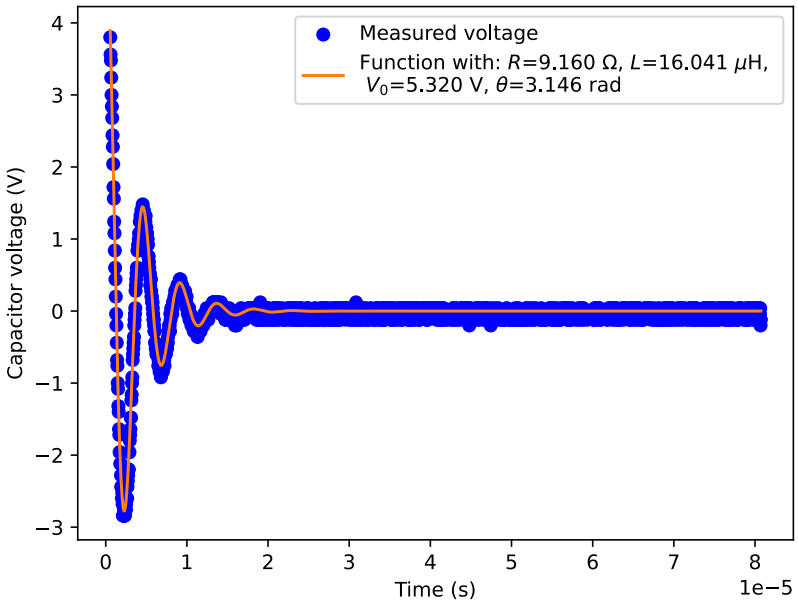

Figure A.6: Function fit with a 32.8 nF capacitor

Appendix B

Verifying the phase resistance of the motor

The datasheet of the motor gives a rating of 85 mΩ [1], however, it does not explain if this is the phase resistance or terminal resistance. Thus, a test must be performed to determine what it means. The setup shown in Figure B.1

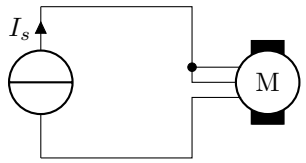

Figure B.1: The measurement setup used to determine the terminal resistance of the motor. I_s is set to 2 A.

is used to determine R, to do this, the voltage is measured directly at the motor terminals, it is important that it is not measured at another location to reduce the effects from the wire resistance. Using this method, a terminal resistance of 65 mΩ was found. As the resistance increases over temperature and frequency, and the measurement was done at DC with a cold motor, it can be expected that the number in the datasheet gives the terminal resistance.

Appendix C

Phase Locked Loop for rotational velocity estimation

For FOC, it has been attempted to implement a Phase Locked Loop (PLL) in order to estimate the angular velocity. However, when implemented in code, the PLL failed to provide useful results. Since implementing the PLL would be an important step if one were to develop FOC further, it has been decided to include the PLL, as currently implemented in the code, in this chapter. While reading this chapter, please keep in mind that this design is flawed or contains a bug. The design method of the PLL is heavily based on the one used in [24].

C.1 Schematic of the PLL

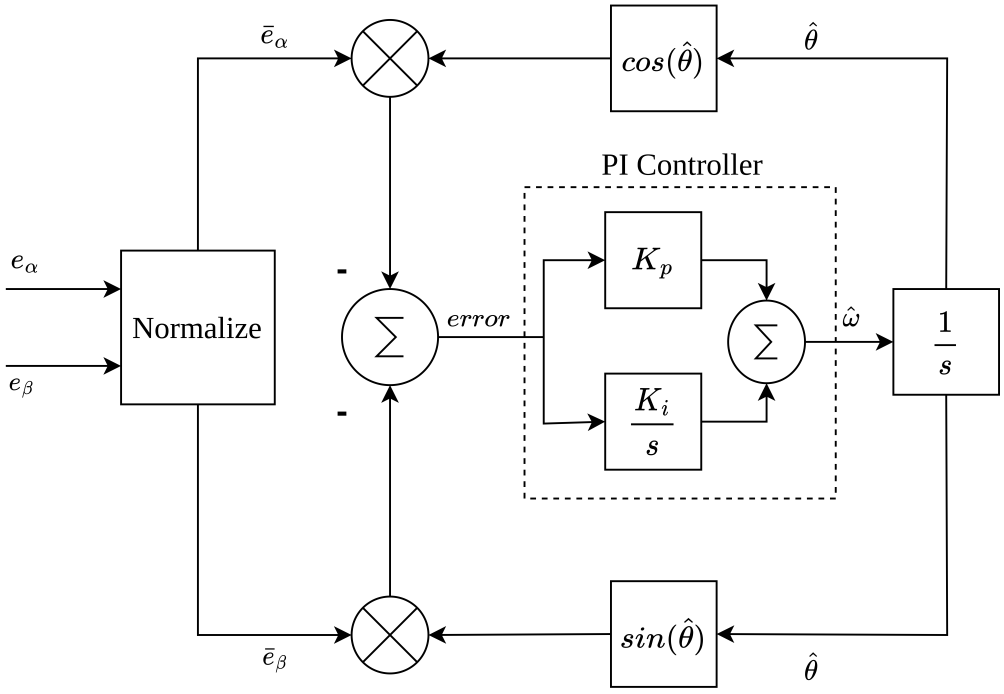

Figure C.1: The schematic for the PLL

The schematic of the PLL is shown in Figure C.1. First, the PLL normalizes the back-emf signals as estimated by the SMO:

$$\bar{e}_{\alpha,\beta} = \frac{e_{\alpha,\beta}}{\sqrt{e_\alpha^2 + e_\beta^2}} \tag{C.1}$$

The normalized back-emf signals can be expressed as:

$$\bar{e}_\alpha = -sin(\theta)\bar{e}_\beta = -cos(\theta) \tag{C.2}$$

Where θ is the rotor angle as estimated by the SMO. Thus, the error signal in the observer is given by:

$$error = sin(\hat{theta})cos(\theta) - cos(\hat{theta})sin(\theta) \tag{C.3}$$

Where $\hat{\theta}$ is the angle as estimated by the PLL. Equation C.3 can be simplified as follows:

$$error = sin(\theta - \hat{\theta}) \tag{C.4}$$

When $\theta - \hat{\theta}$ is small, Equation C.4 can be further simplified to:

$$error = \theta - \hat{\theta} \tag{C.5}$$

This error is then fed to a PI controller, which estimates the angular velocity.

C.2 Tuning the PI controller

The function of the estimated angle by the PLL is given by

$$\hat{\theta} = (\theta - \hat{\theta})(K_p + \frac{K_i}{s})\frac{1}{s} \tag{C.6}$$

The transfer function of the PLL is thus given by:

$$\frac{\hat{\theta}}{\theta} = \frac{K_p s + K_i}{s^2 + K_p s + K_i} \tag{C.7}$$

In this equation, the damping factor is given as follows:

$$\zeta = \frac{K_p}{4K_i} \tag{C.8}$$

In order for this system to be critically damped, the damping factor should be set to 1. Thus, the following relation is obtained:

$$\begin{aligned} K_i &= \frac{K_p^2}{4} \\ K_p &= 2cK_i = c^2 \end{aligned} \tag{C.9}$$

In order to find the cutoff frequency in rad, Equation C.7 needs to be rewritten, and the following equation needs to be solved for s:

$$\frac{2cs + c^2}{s^2 + 2cs + c^2} = \frac{G(0)}{\sqrt{2}} \tag{C.10}$$

Where $G(0)$ is the zero frequency gain. As the zero frequency gain is unity, the equation above can be simplified. The expression for s is now given as:

$$s = c(\sqrt{2 - \sqrt{2}} + \sqrt{2} - 1) \tag{C.11}$$

Since the cutoff frequency should be at least equal to $14\,661\,\mathrm{rad\,s^{-1}}$, which is the maximum speed, c should be equal to:

$$c = 12429 \tag{C.12}$$

Appendix D

V/f simulations

D.1 No Rate Limiter

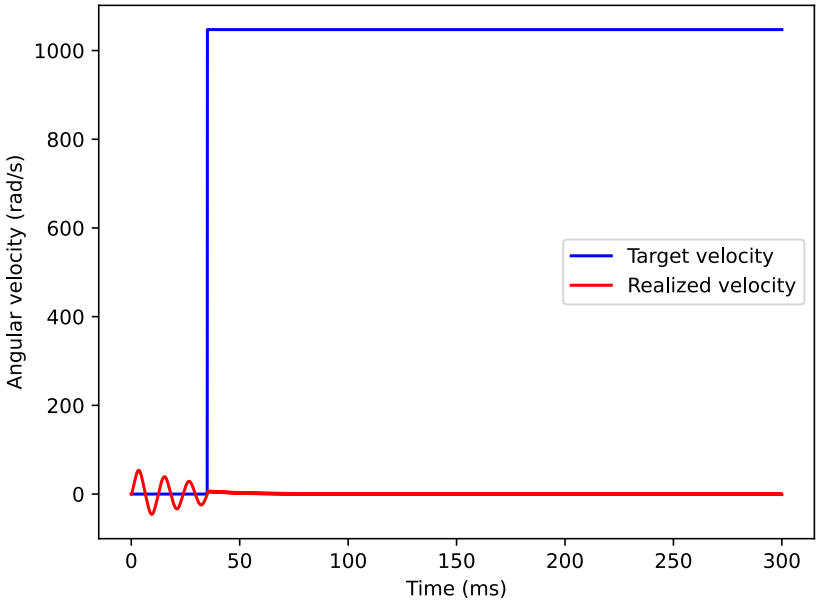

Figure D.1: The desired and realized velocity

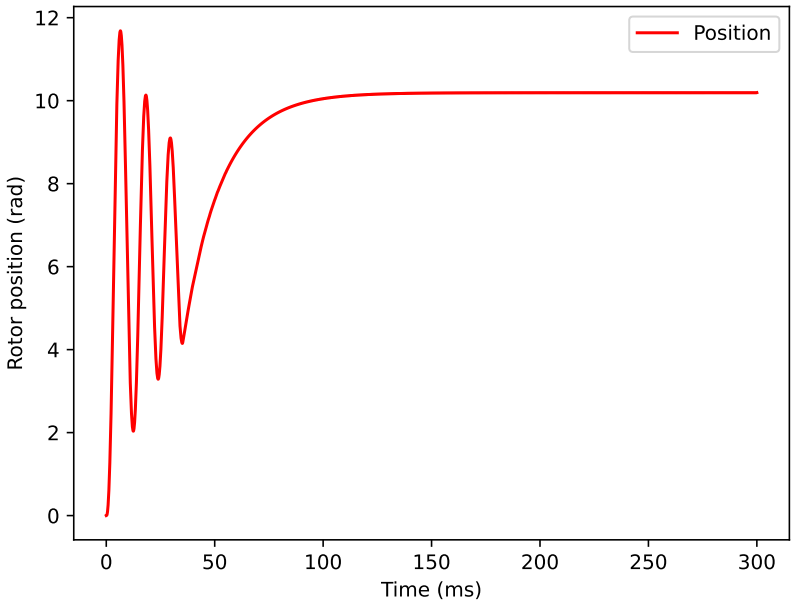

Figure D.2: The position of the rotor

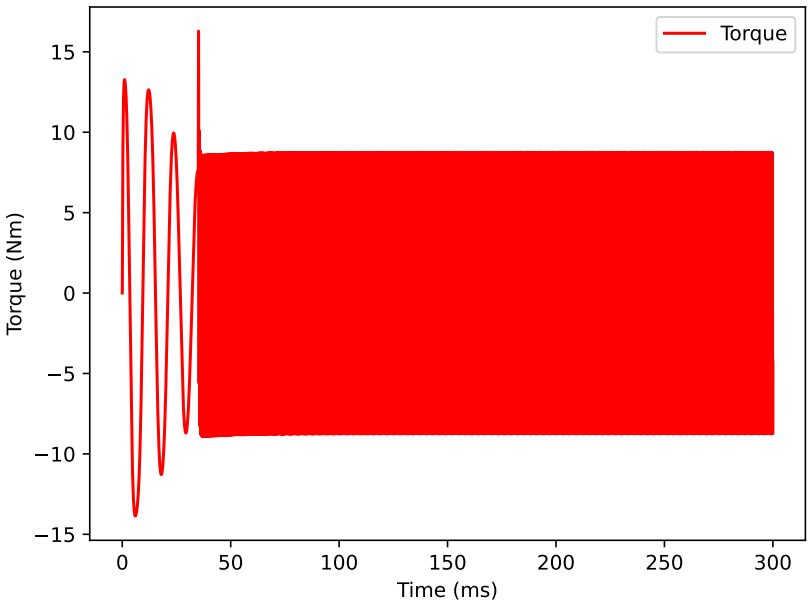

Figure D.3: The torque delivered by the motor

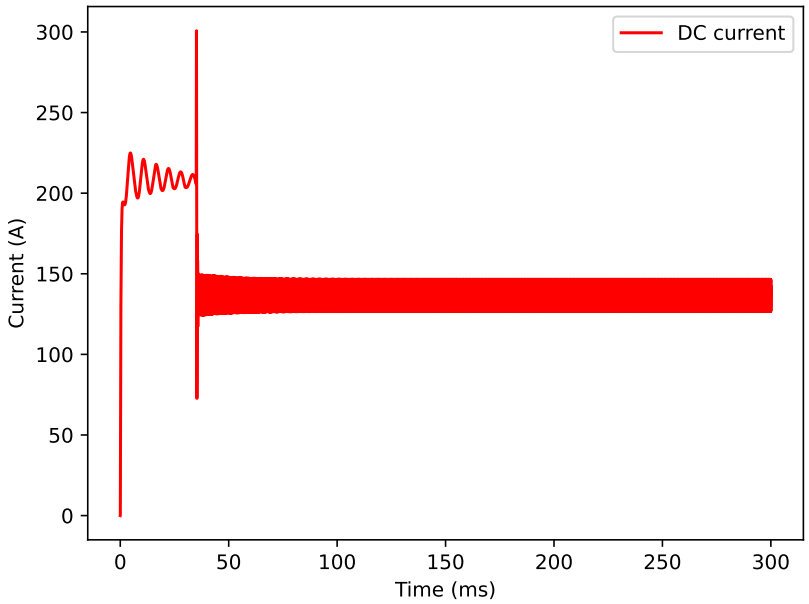

Figure D.4: The DC current through the motor

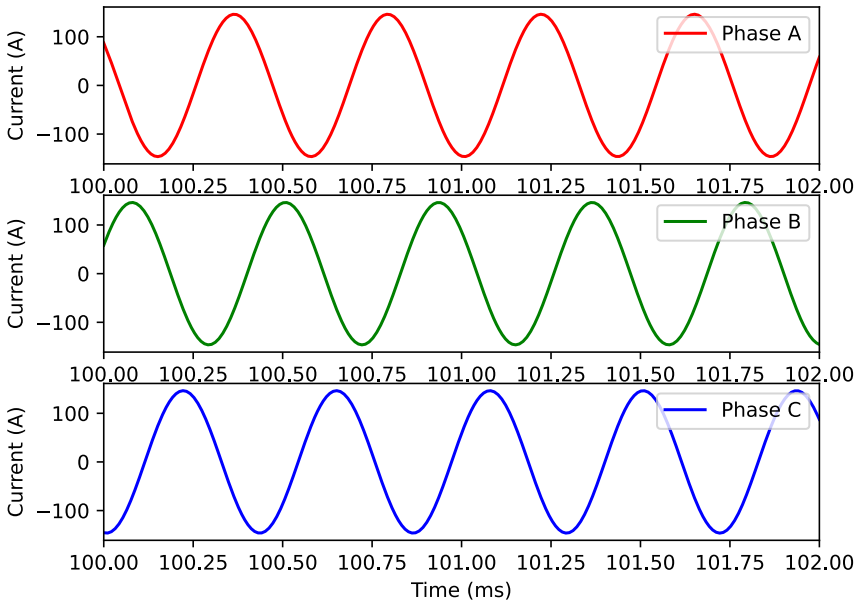

Figure D.5: Zoomed in phase currents through the motor

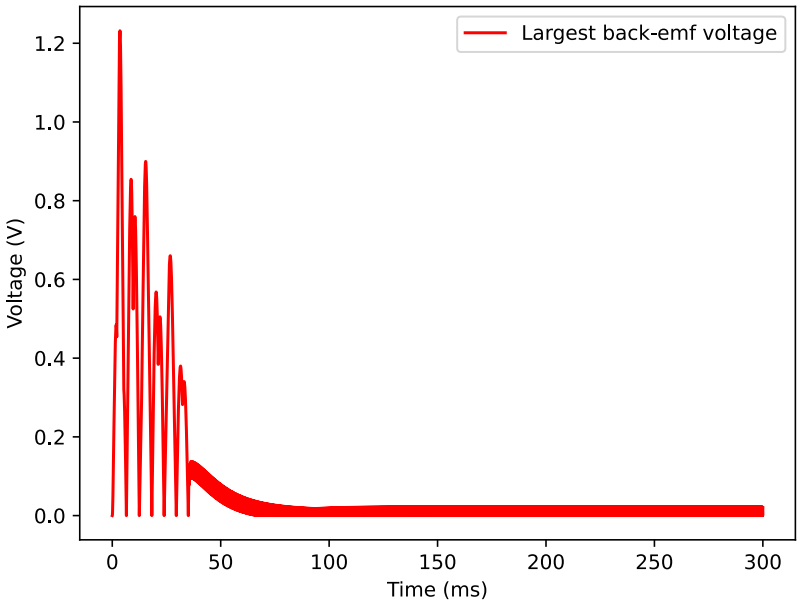

Figure D.6: The largest value for back-emf generated by a motor phase

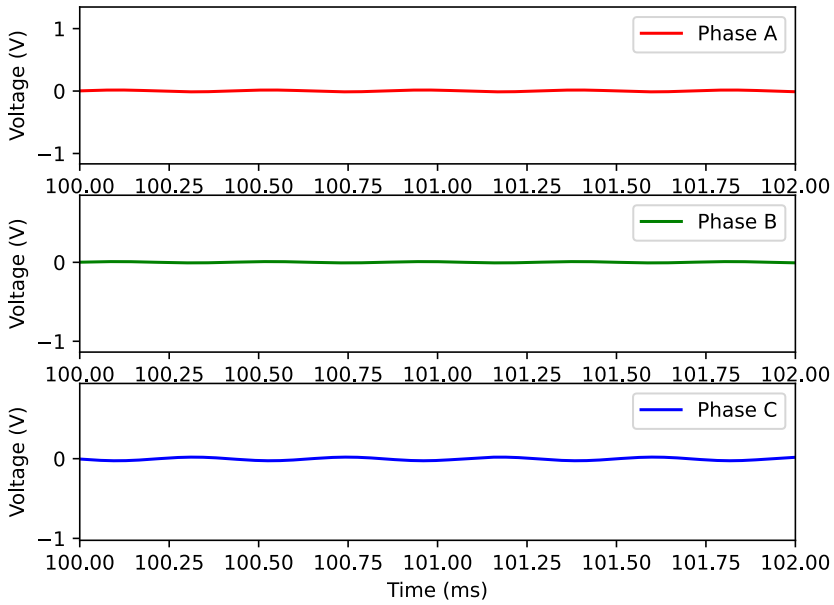

Figure D.7: Zoomed in back-emf generated by each phase

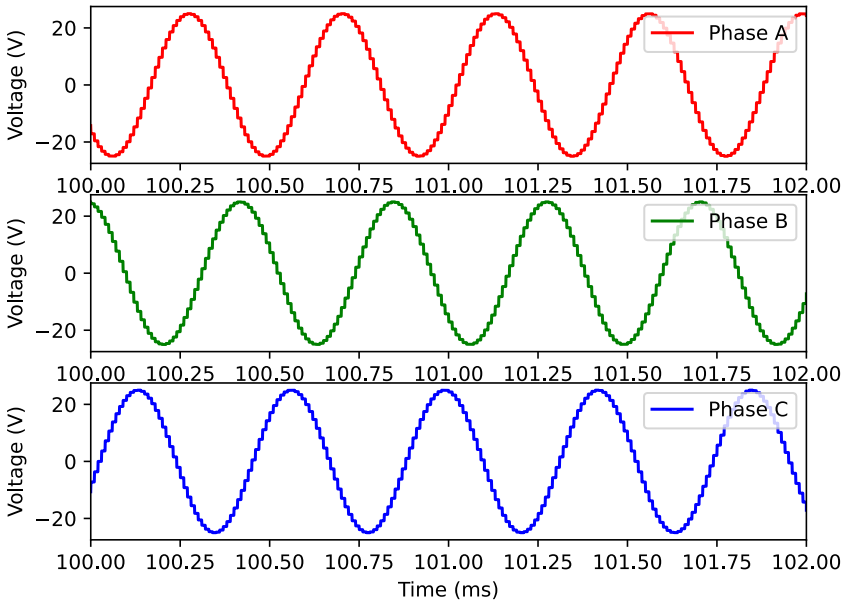

Figure D.8: Zoomed in phase voltages

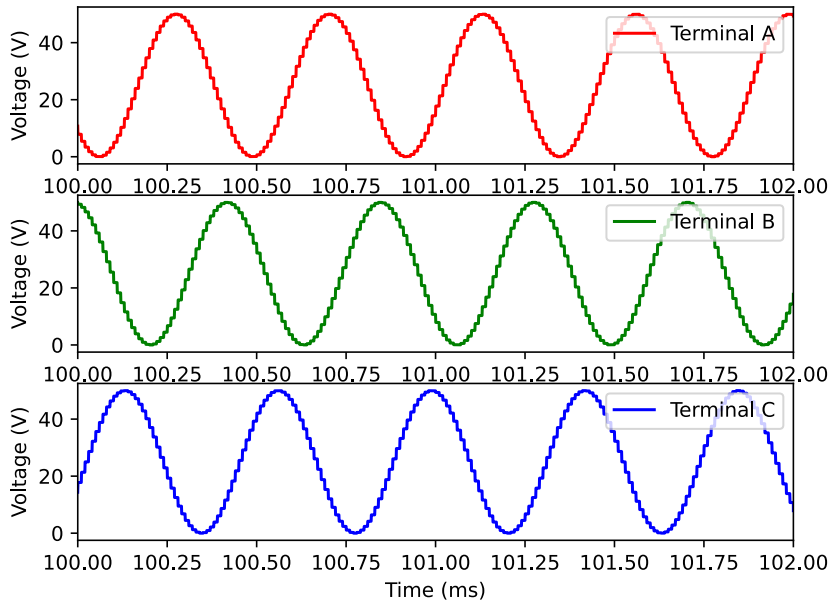

Figure D.9: Zoomed in terminal voltages

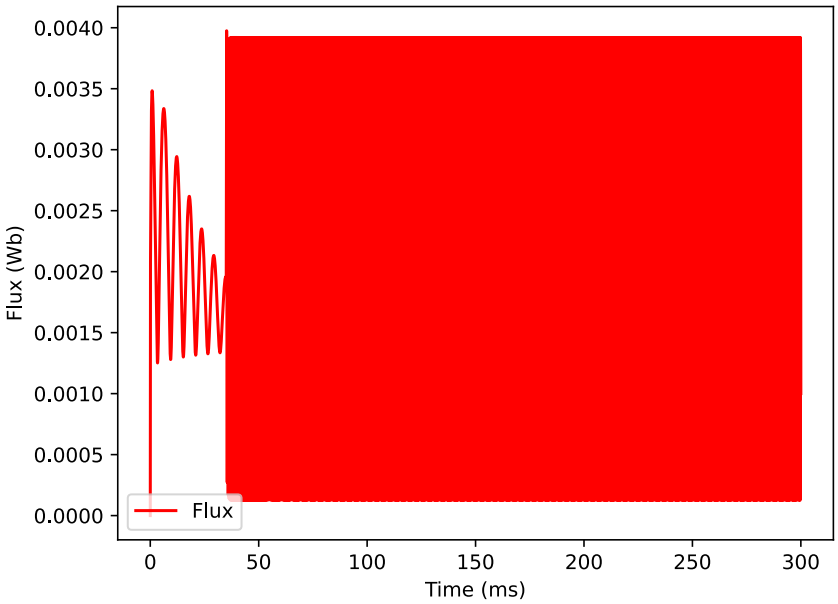

Figure D.10: Flux generated by the motor, for more information see code

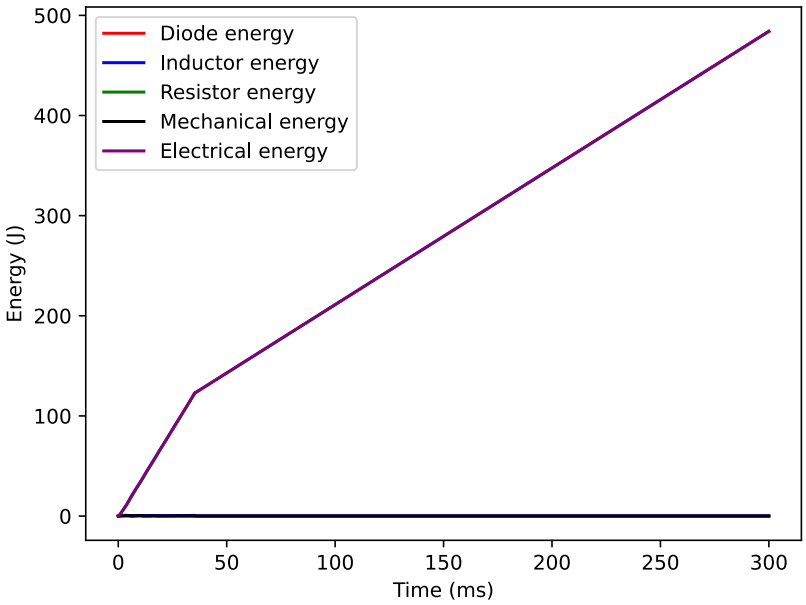

Figure D.11: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.000

D.2 With Rate Limiter

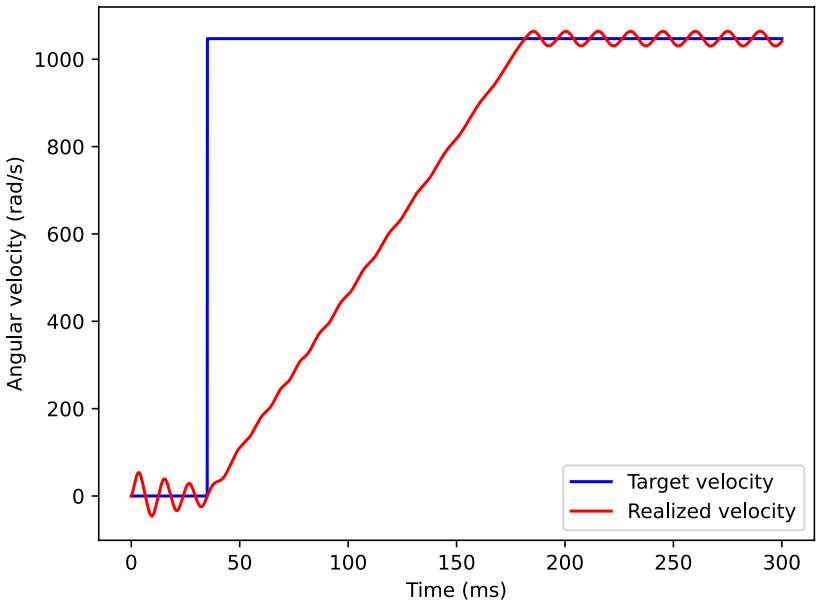

Figure D.12: The desired and realized velocity

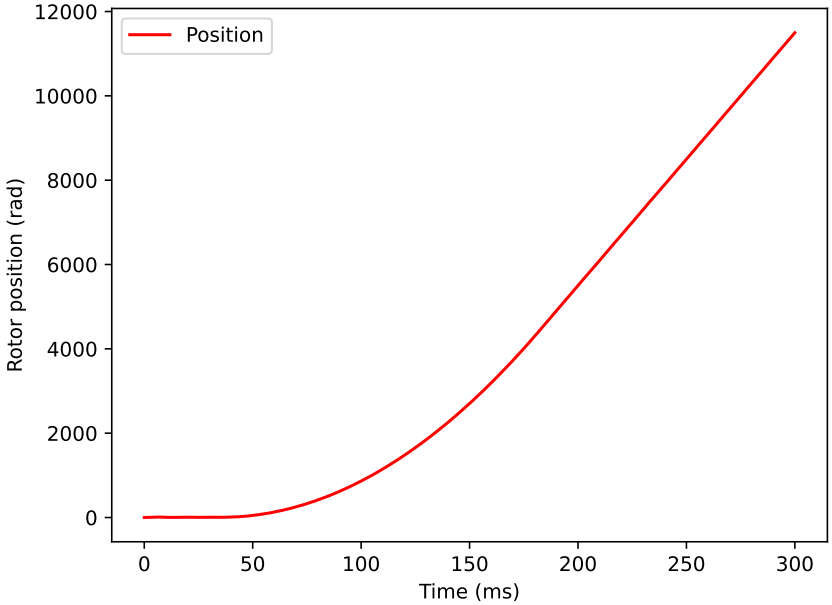

Figure D.13: The position of the rotor

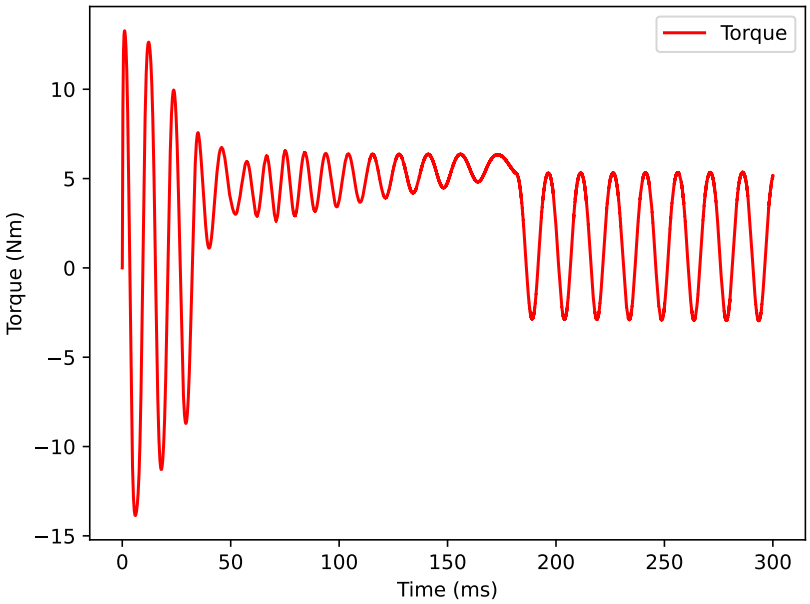

Figure D.14: The torque delivered by the motor

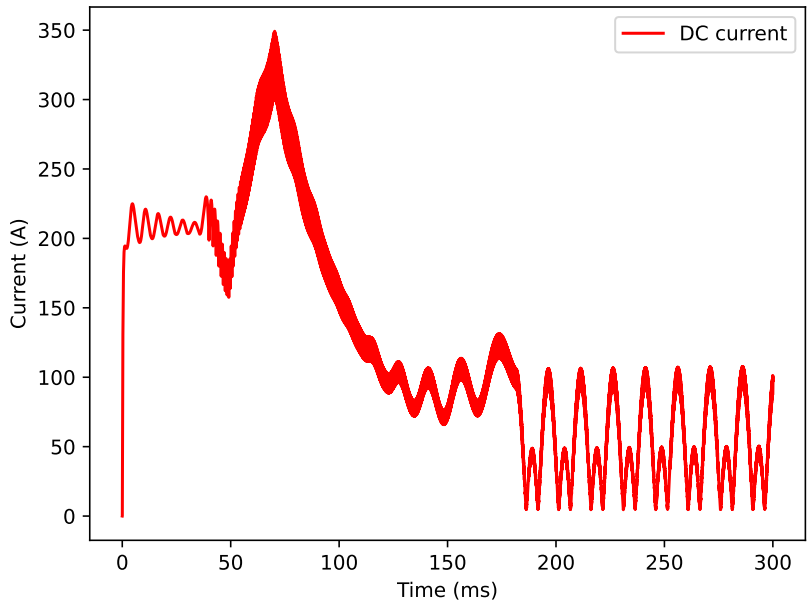

Figure D.15: The DC current through the motor

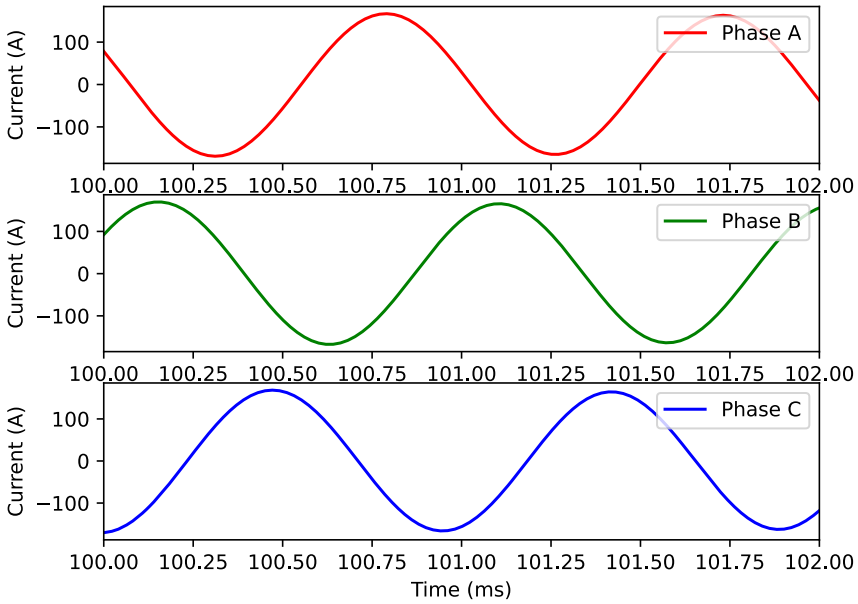

Figure D.16: Zoomed in phase currents through the motor

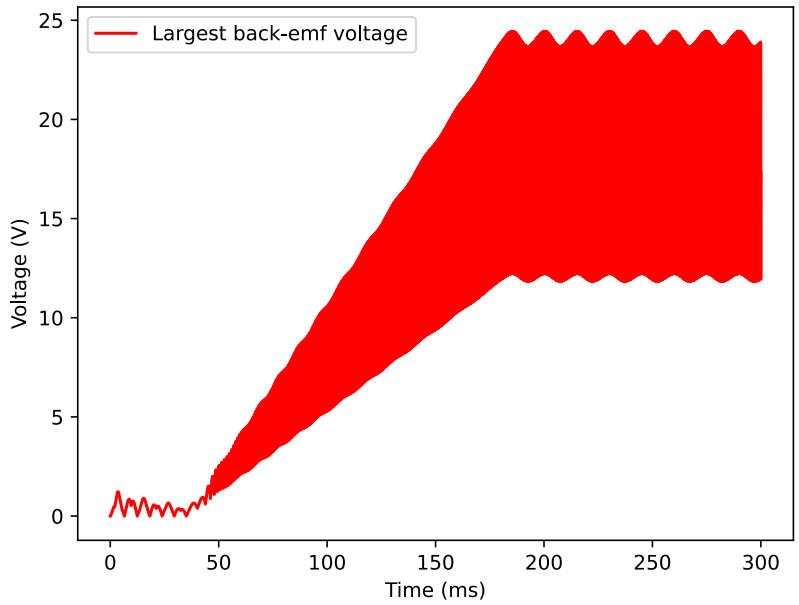

Figure D.17: The largest value for back-emf generated by a motor phase

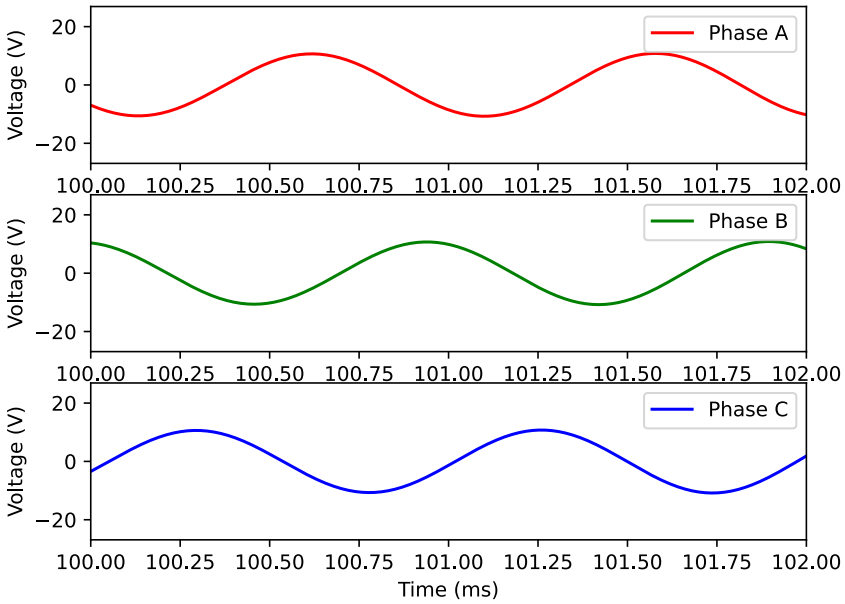

Figure D.18: Zoomed in back-emf generated by each phase

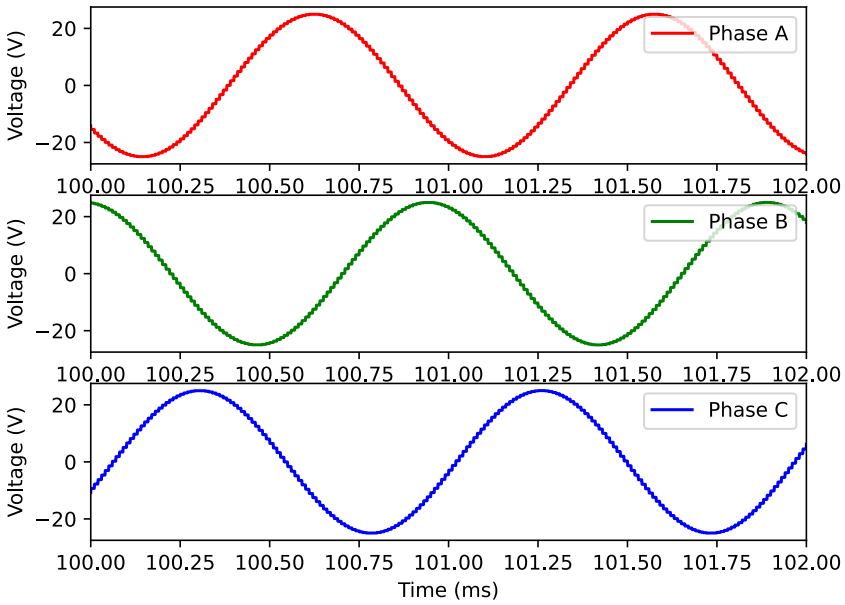

Figure D.19: Zoomed in phase voltages

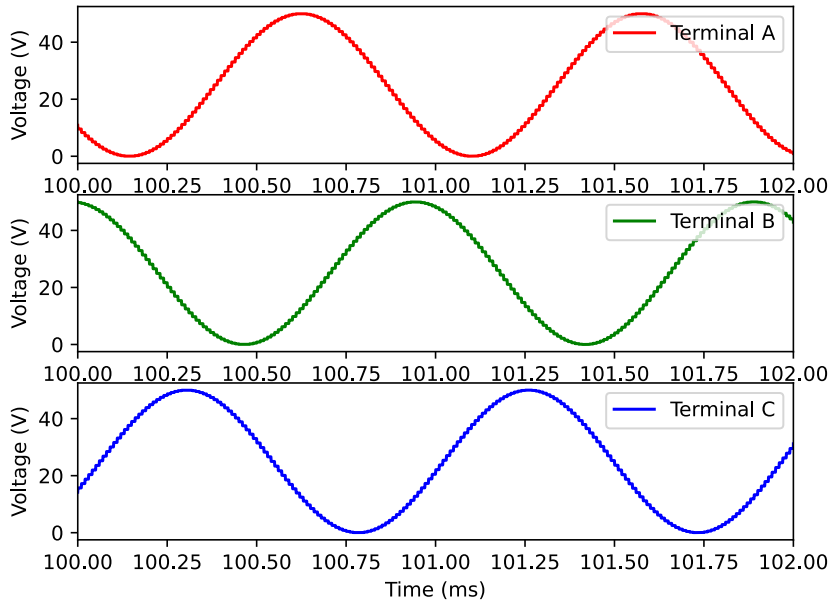

Figure D.20: Zoomed in terminal voltages

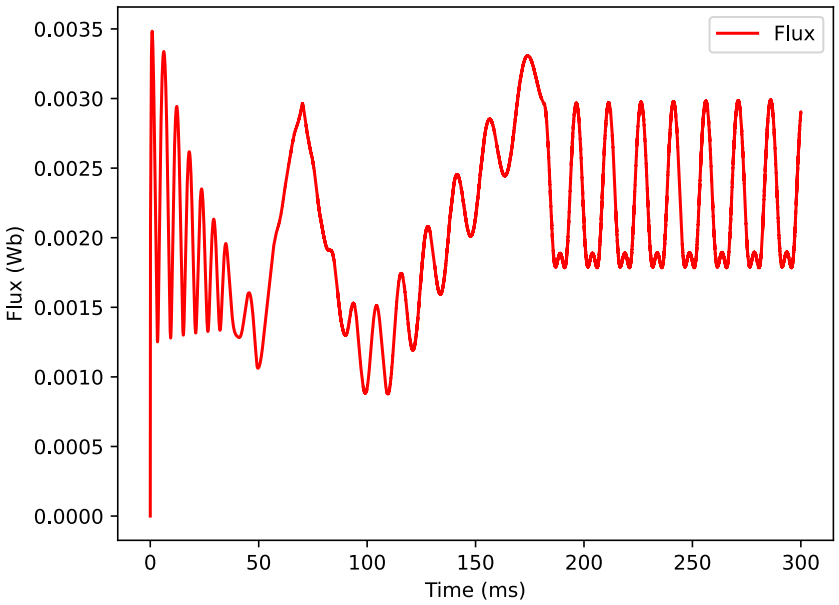

Figure D.21: Flux generated by the motor, for more information see code

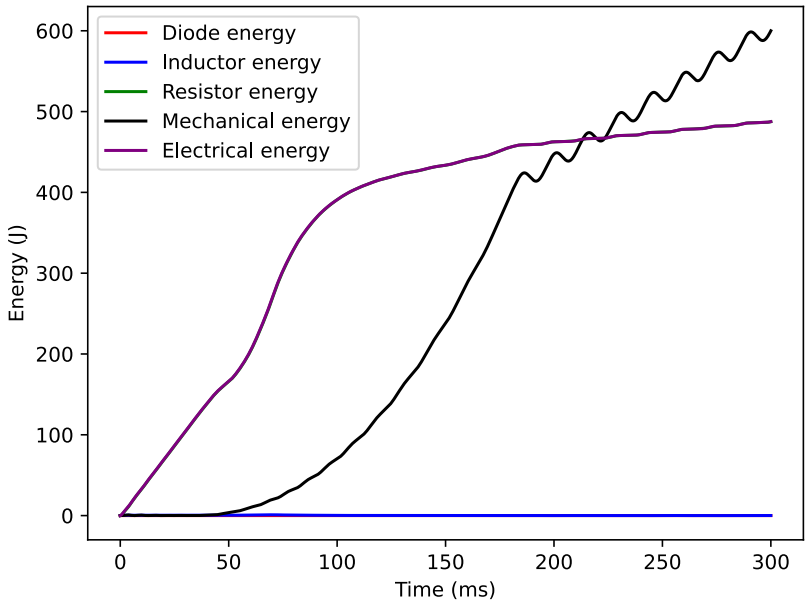

Figure D.22: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.650

D.3 Unreachable Desired Speed

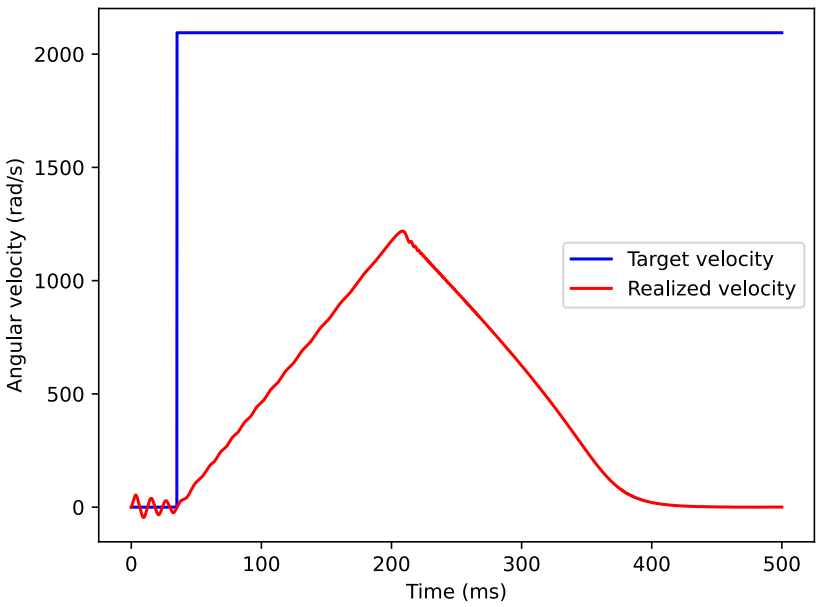

Figure D.23: The desired and realized velocity

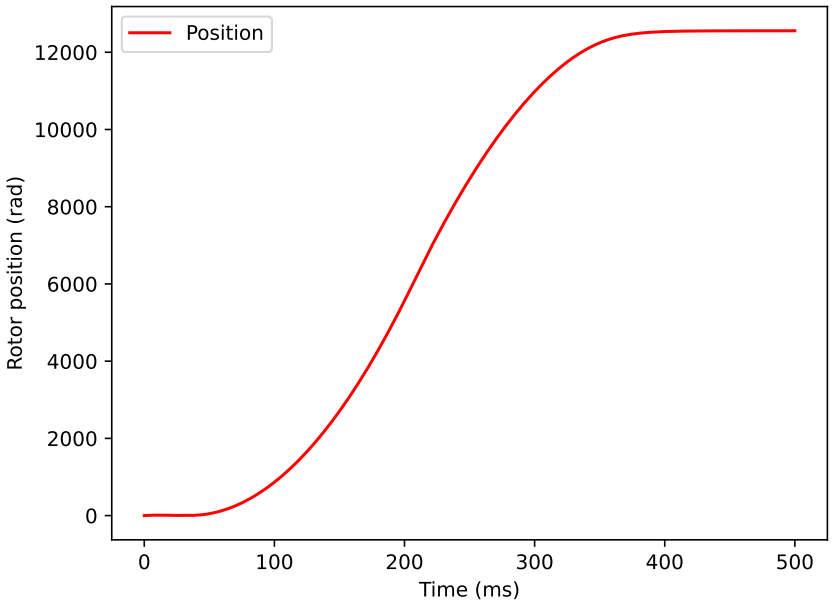

Figure D.24: The position of the rotor

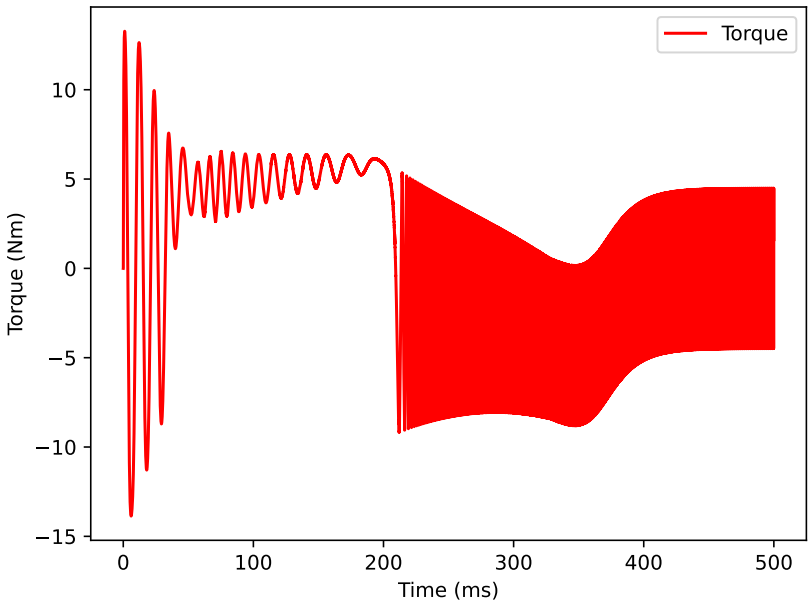

Figure D.25: The torque delivered by the motor

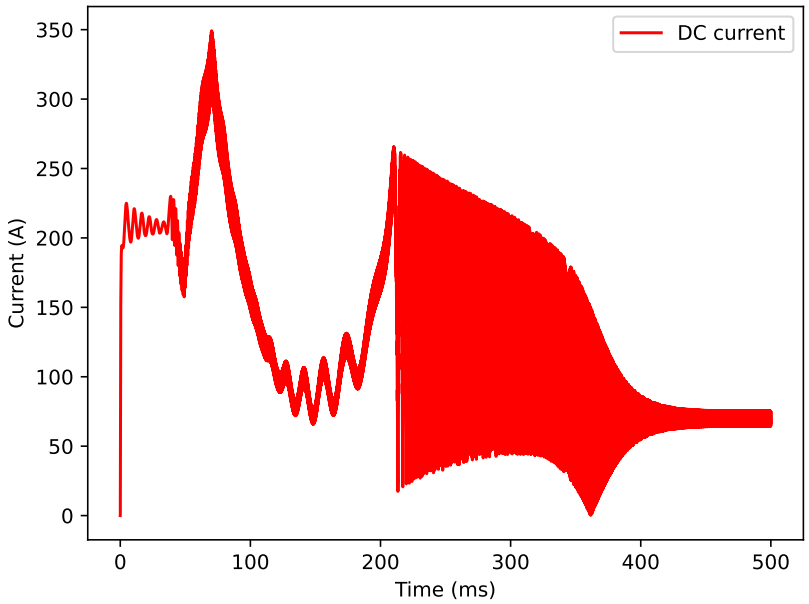

Figure D.26: The DC current through the motor

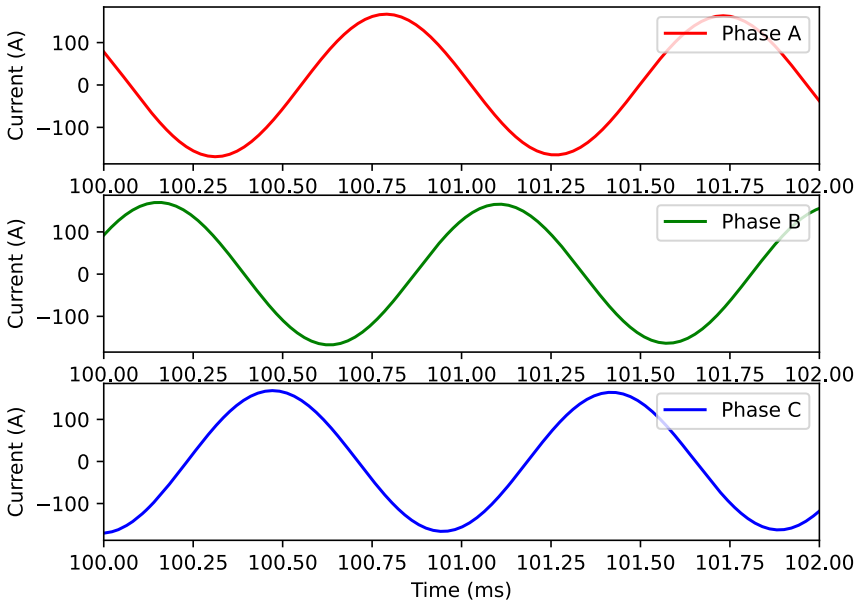

Figure D.27: Zoomed in phase currents through the motor

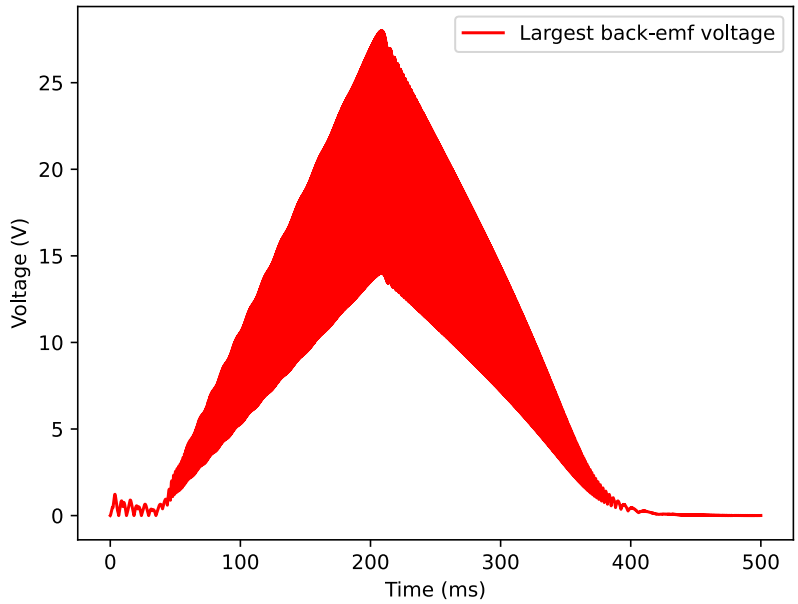

Figure D.28: The largest value for back-emf generated by a motor phase

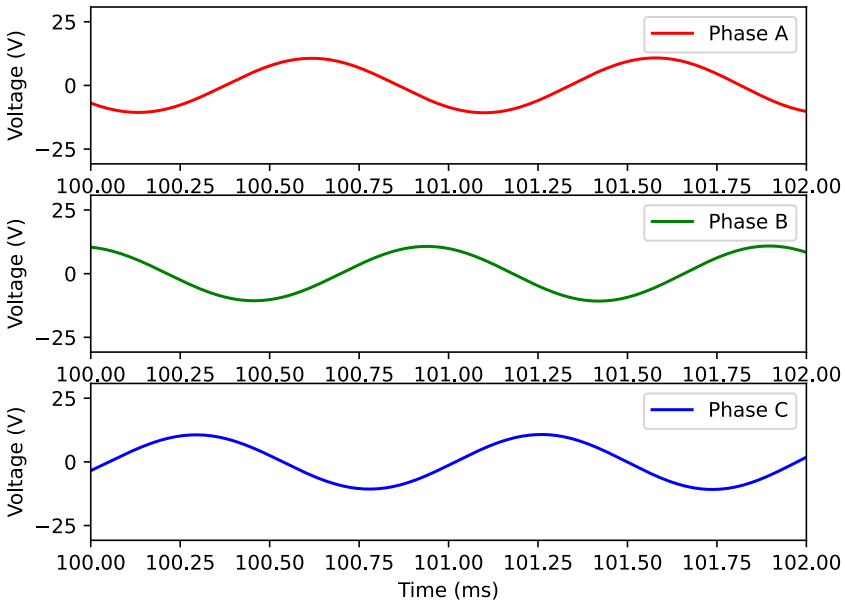

Figure D.29: Zoomed in back-emf generated by each phase

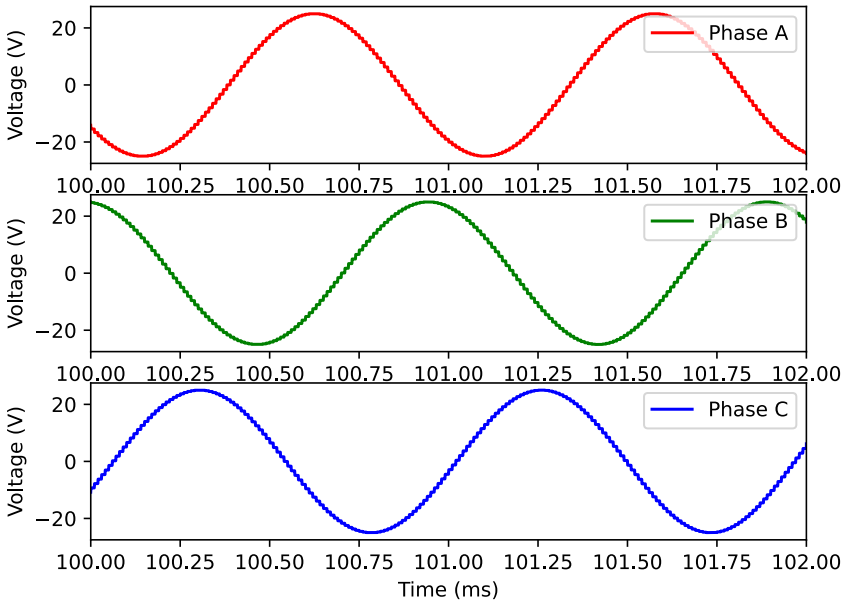

Figure D.30: Zoomed in phase voltages

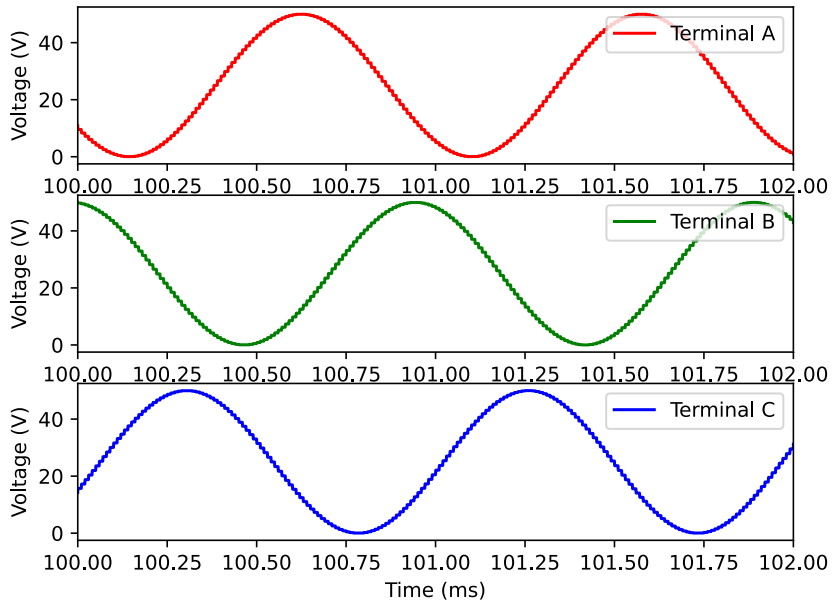

Figure D.31: Zoomed in terminal voltages

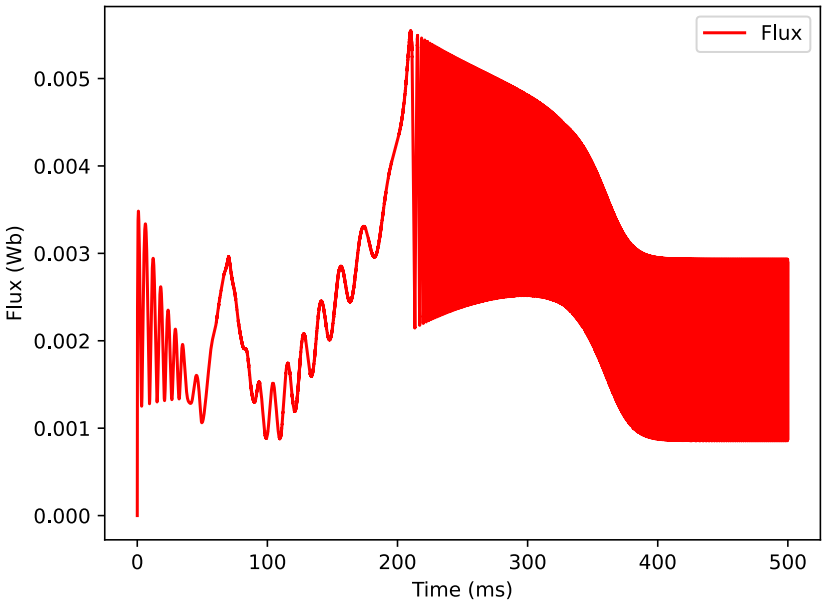

Figure D.32: Flux generated by the motor, for more information see code

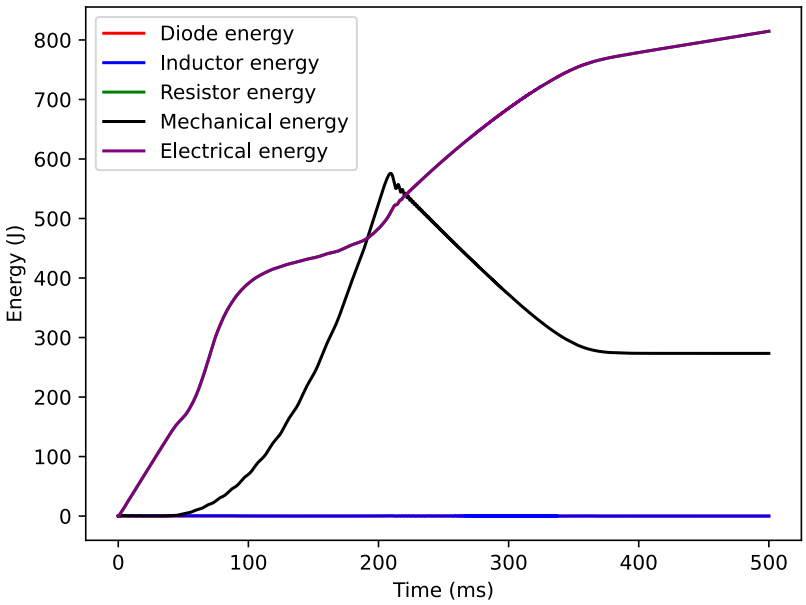

Figure D.33: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.425

Appendix E

First implementation of FOC simulations

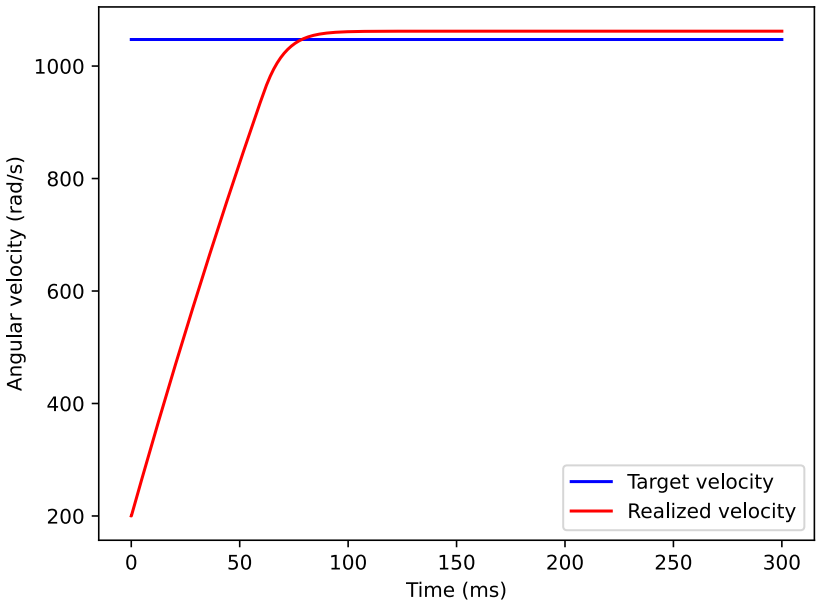

Figure E.1: The desired and realized velocity

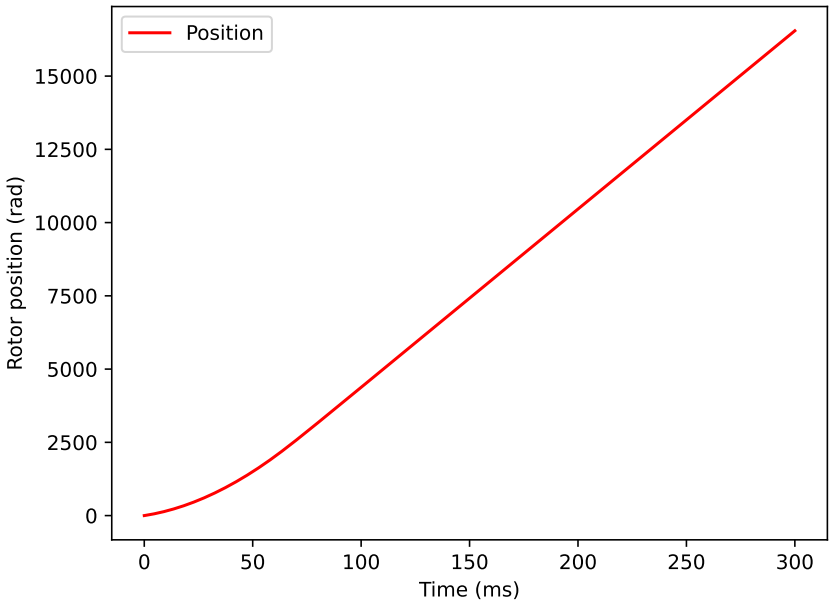

Figure E.2: The position of the rotor

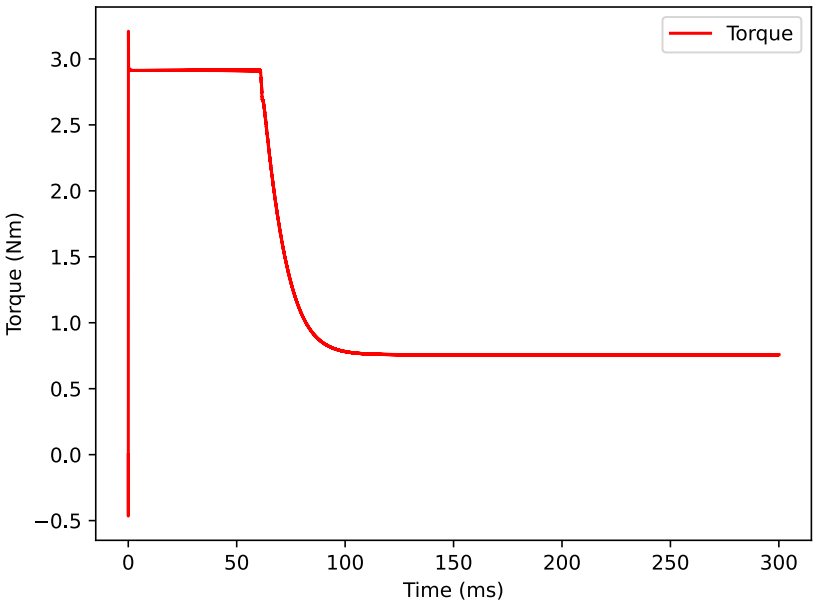

Figure E.3: The torque delivered by the motor

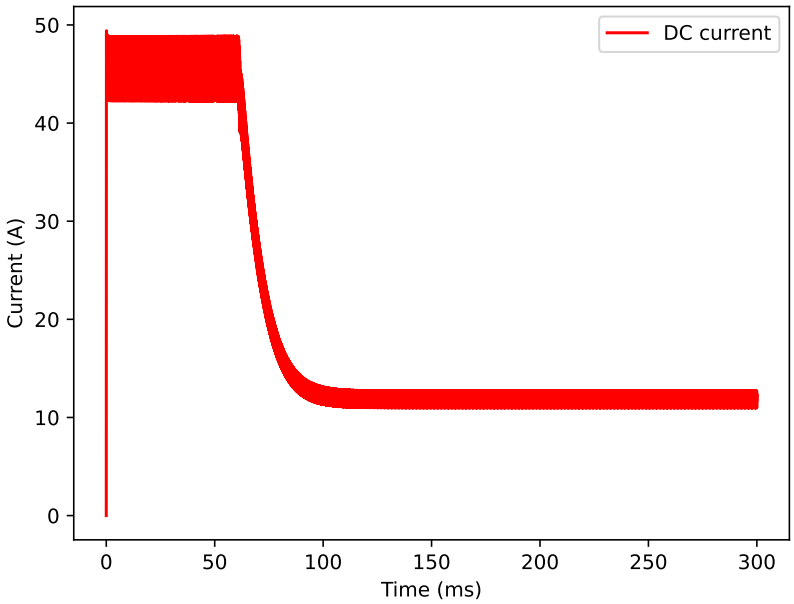

Figure E.4: The DC current through the motor

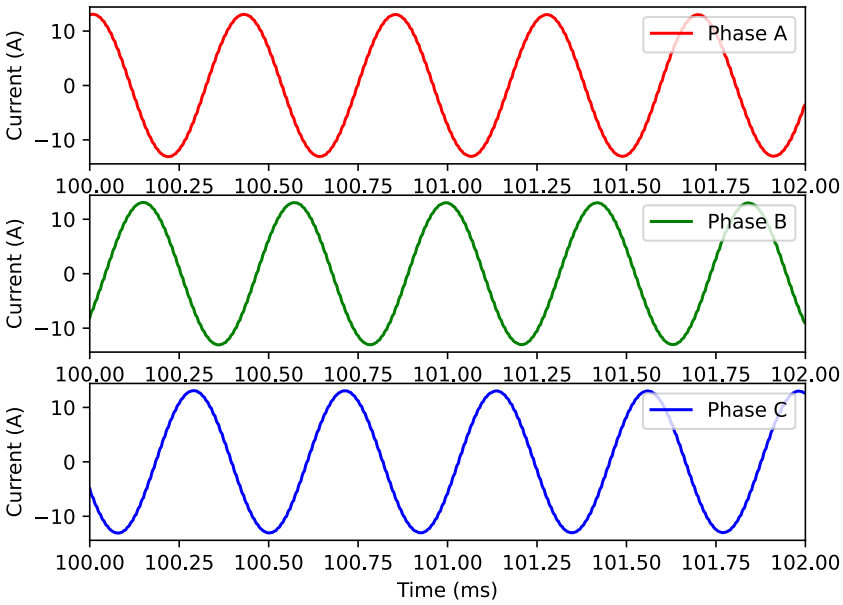

Figure E.5: Zoomed in phase currents through the motor

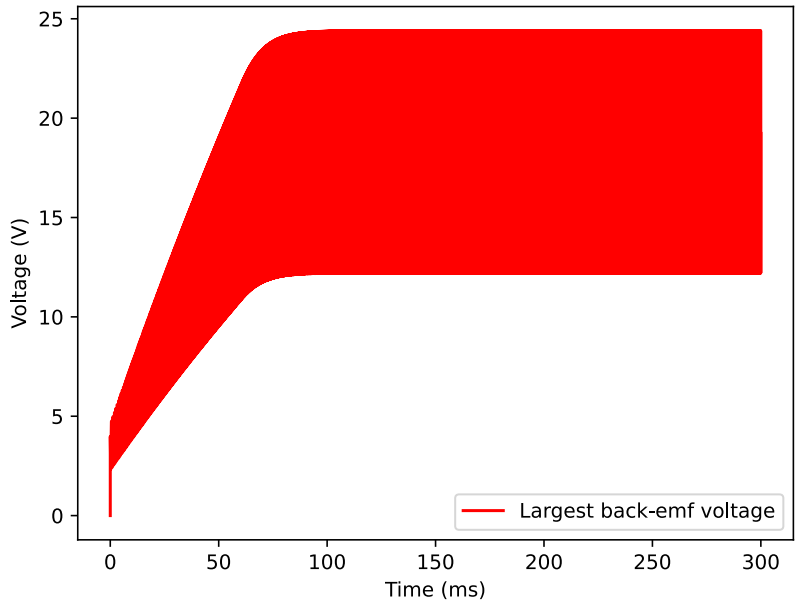

Figure E.6: The largest value for back-emf generated by a motor phase

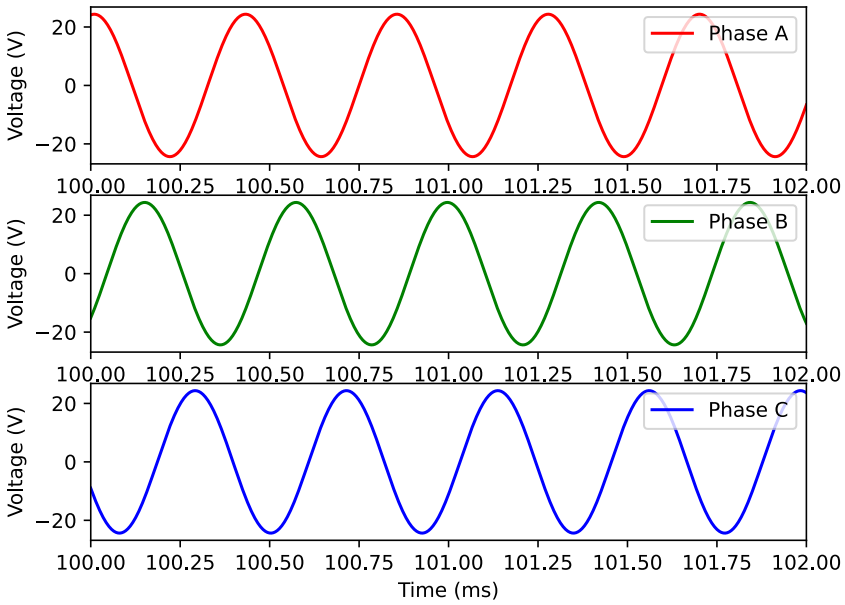

Figure E.7: Zoomed in back-emf generated by each phase

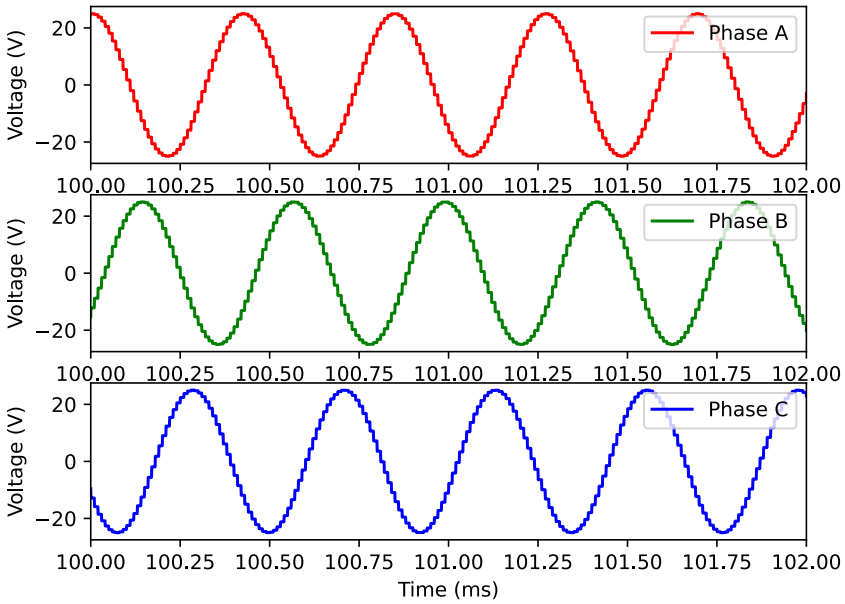

Figure E.8: Zoomed in phase voltages

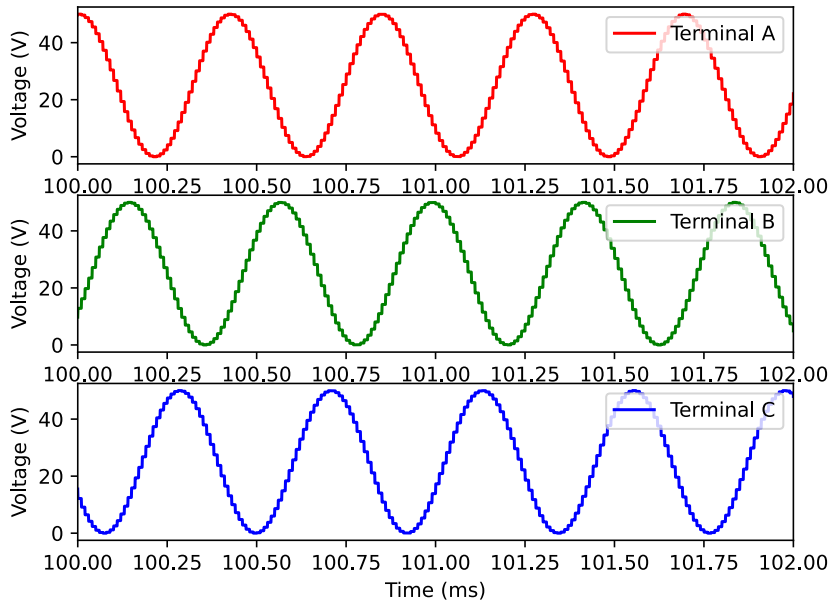

Figure E.9: Zoomed in terminal voltages

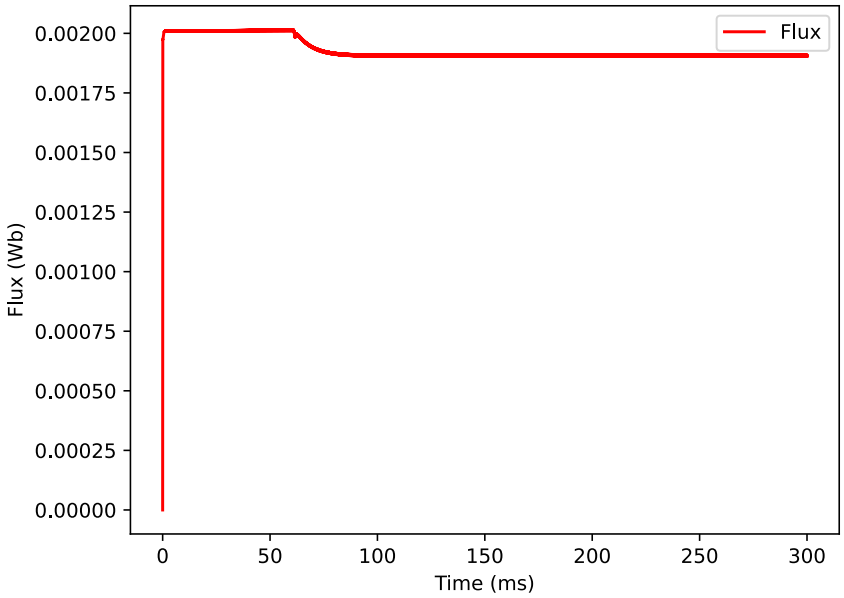

Figure E.10: Flux generated by the motor, for more information see code

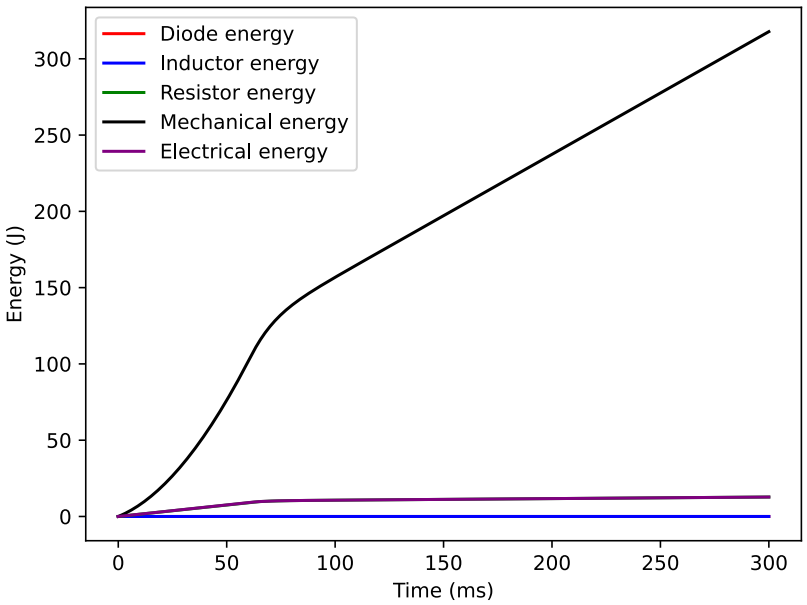

Figure E.11: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.961

Appendix F

First implementation of trapezoidal control simulations

As no current limiter was implemented, the peak currents are very high, be wary about this when reading these plots. Also, the PI controller has not been fully tuned.

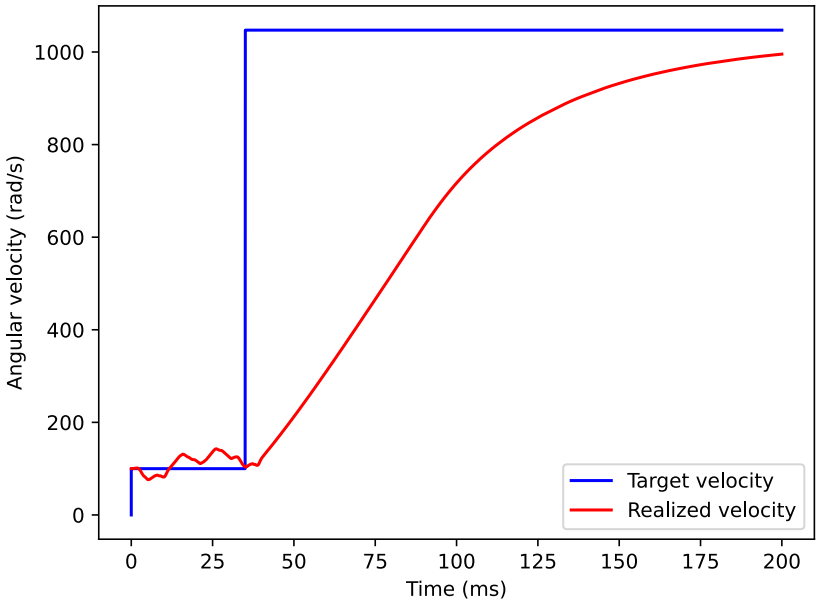

Figure F.1: The desired and realized velocity

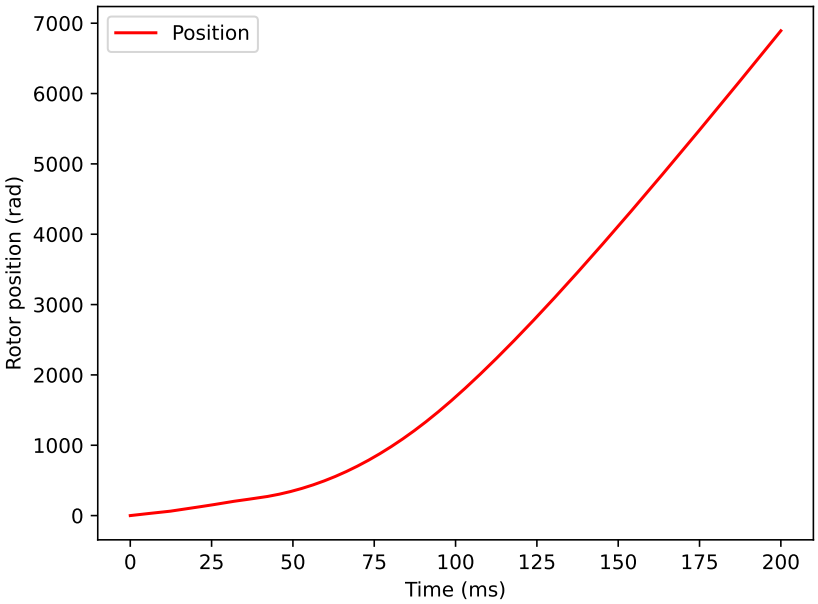

Figure F.2: The position of the rotor

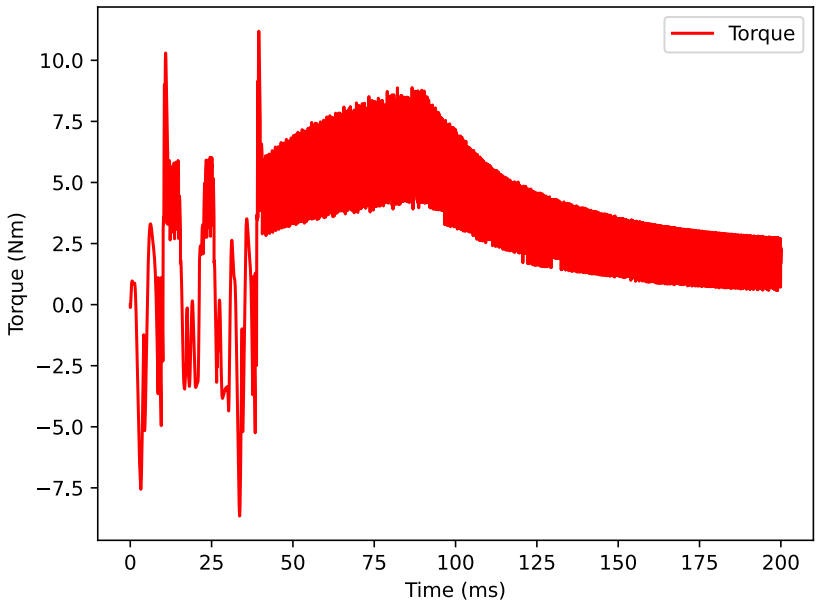

Figure F.3: The torque delivered by the motor

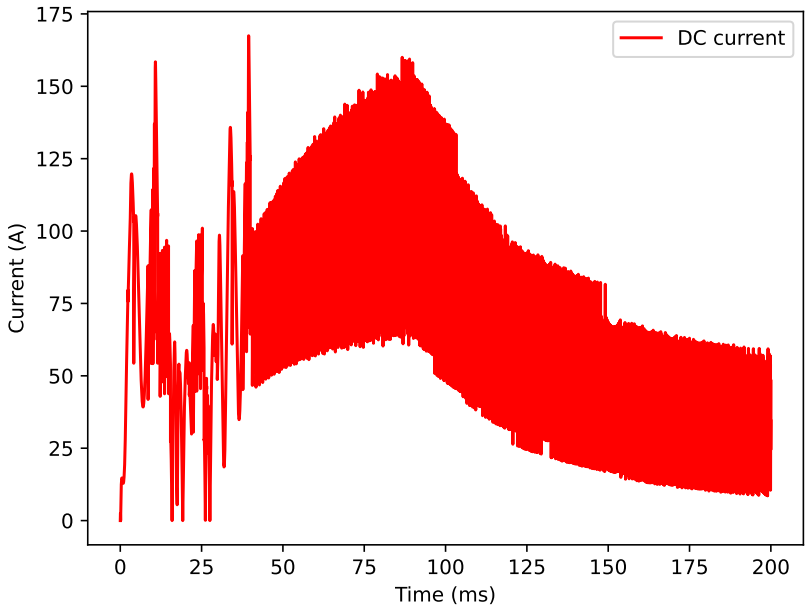

Figure F.4: The DC current through the motor

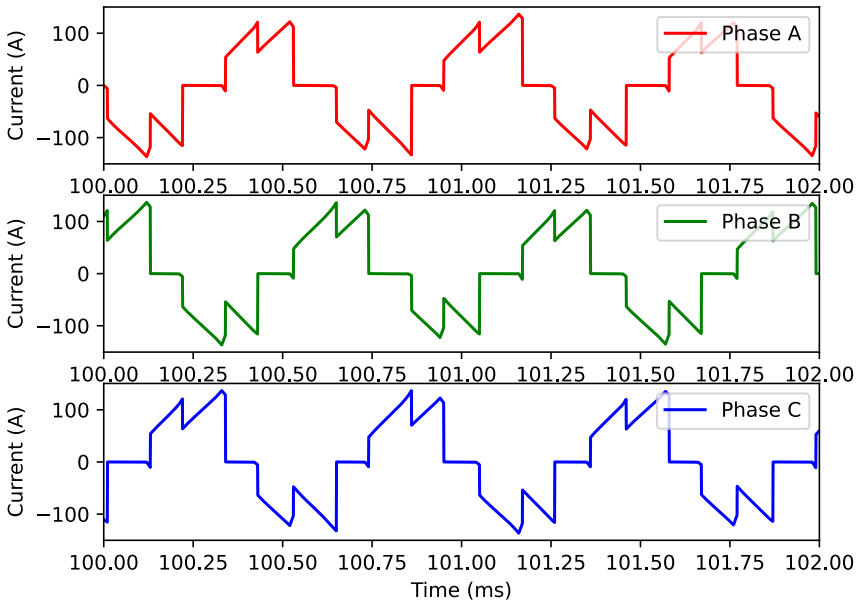

Figure F.5: Zoomed in phase currents through the motor

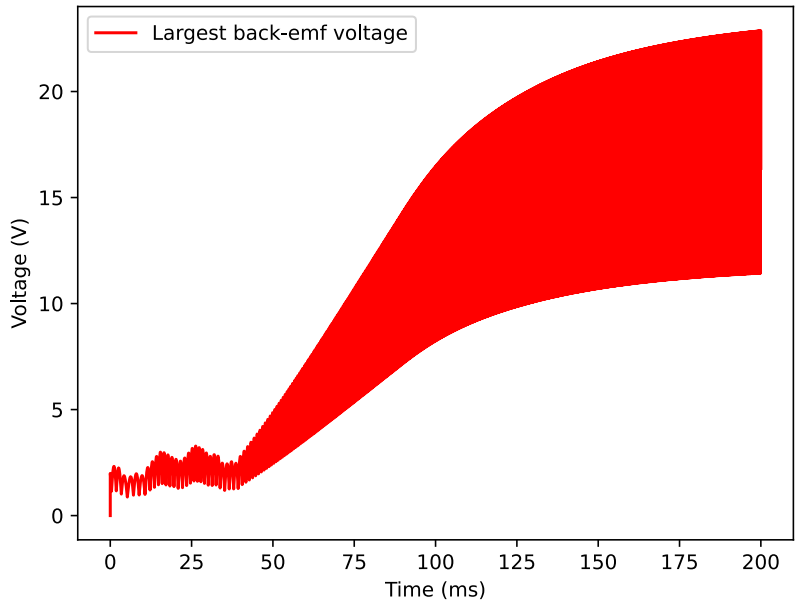

Figure F.6: The largest value for back-emf generated by a motor phase

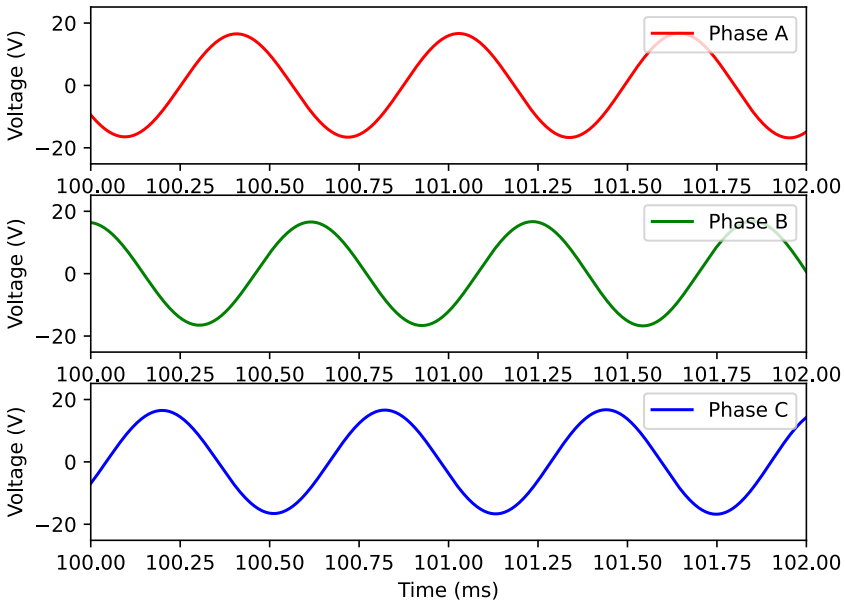

Figure F.7: Zoomed in back-emf generated by each phase

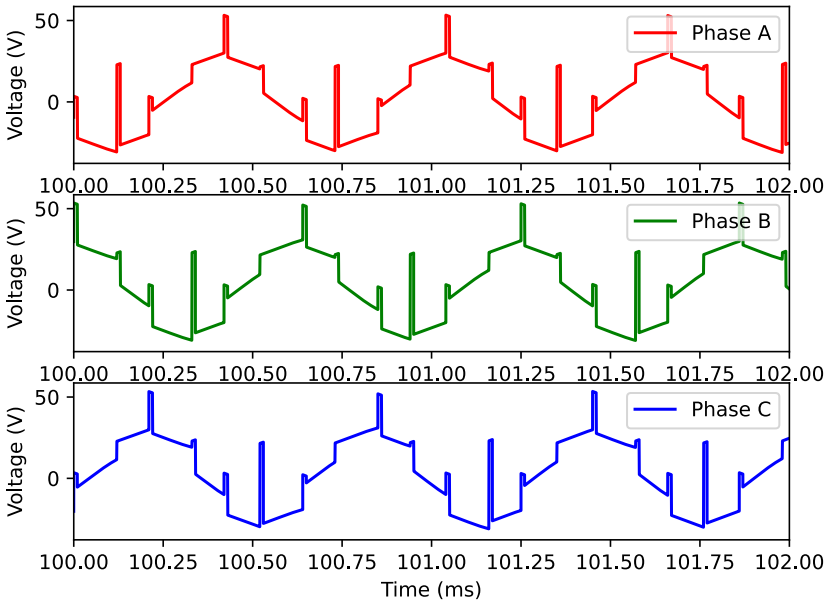

Figure F.8: Zoomed in phase voltages

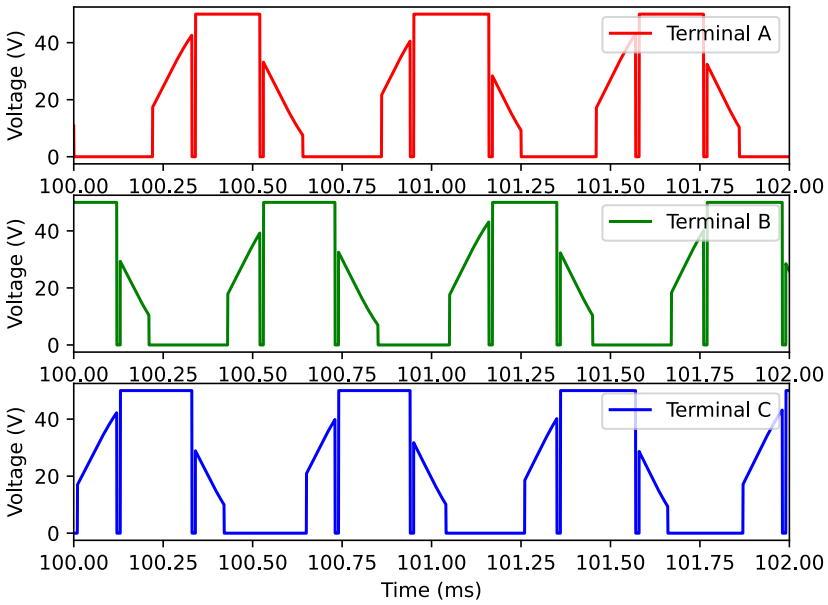

Figure F.9: Zoomed in terminal voltages

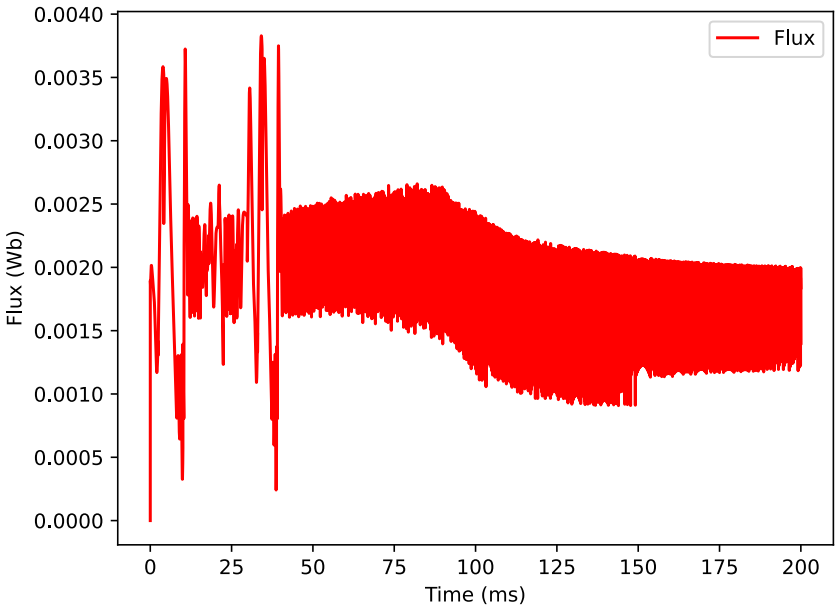

Figure F.10: Flux generated by the motor, for more information see code

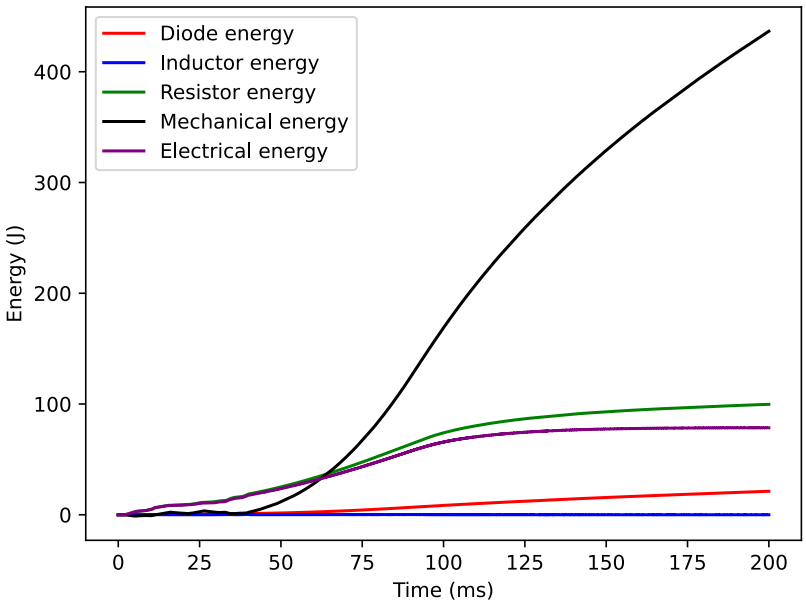

Figure F.11: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.840

Appendix G

Simulations for trapezoidal control tuning

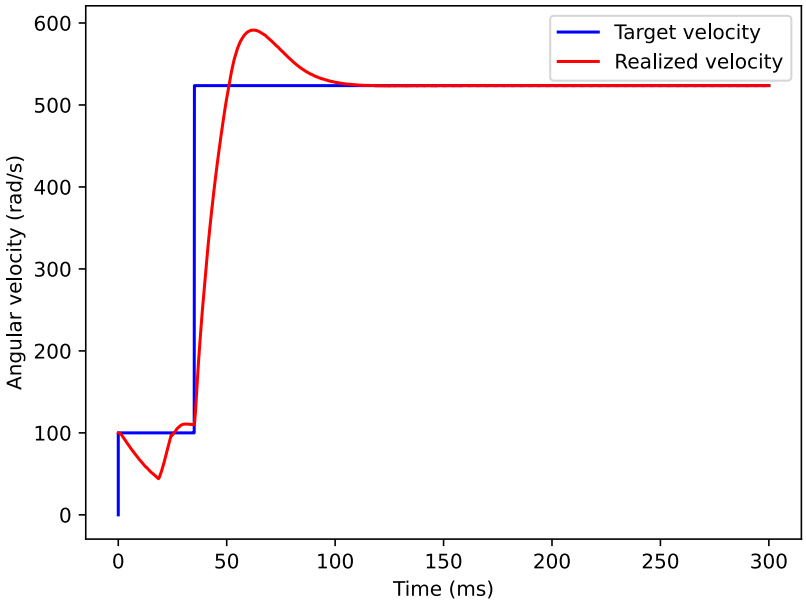

Figure G.1: The desired and realized velocity

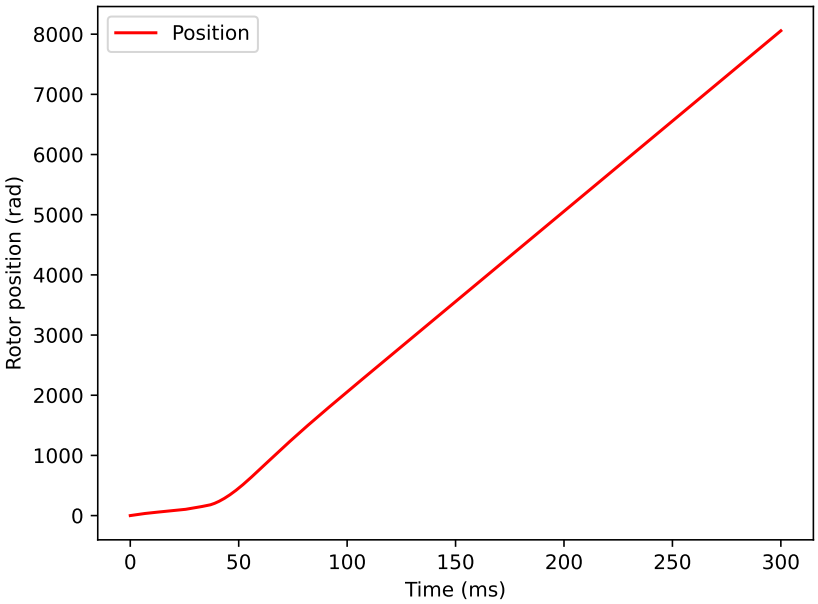

Figure G.2: The position of the rotor

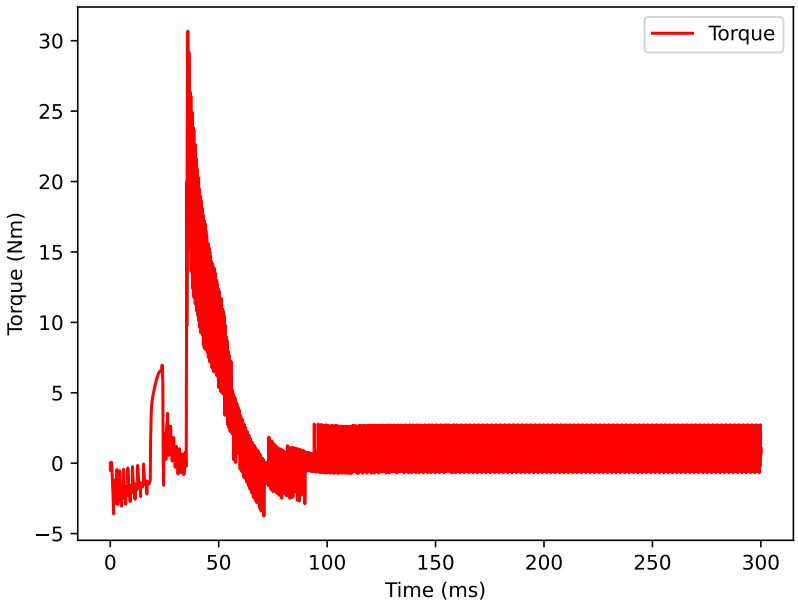

Figure G.3: The torque delivered by the motor

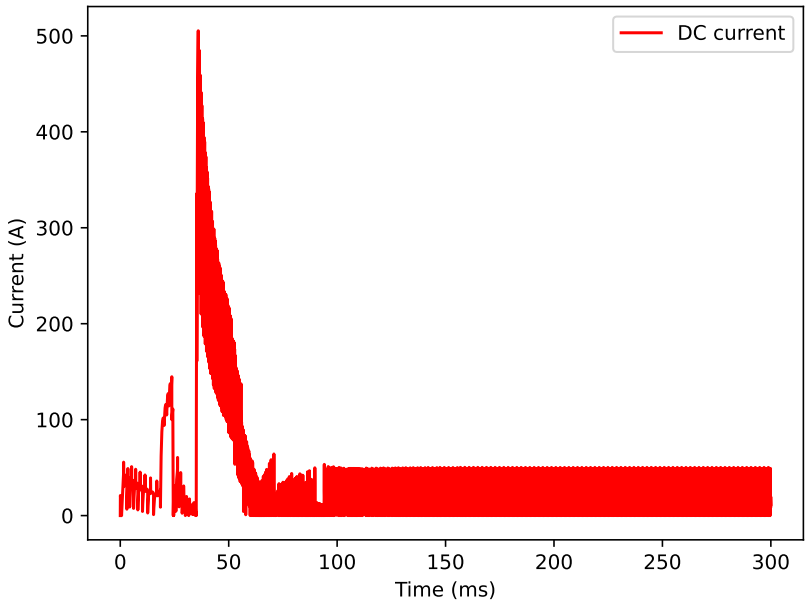

Figure G.4: The DC current through the motor

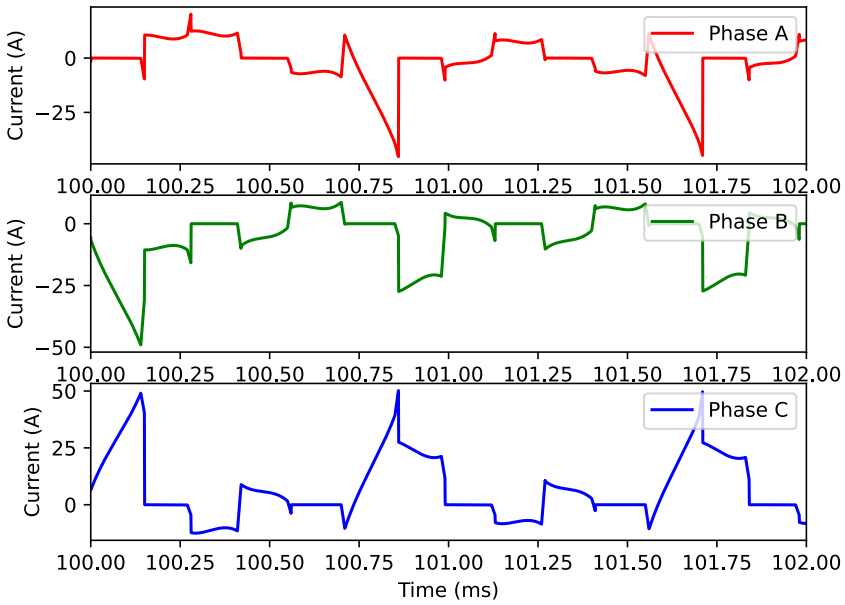

Figure G.5: Zoomed in phase currents through the motor

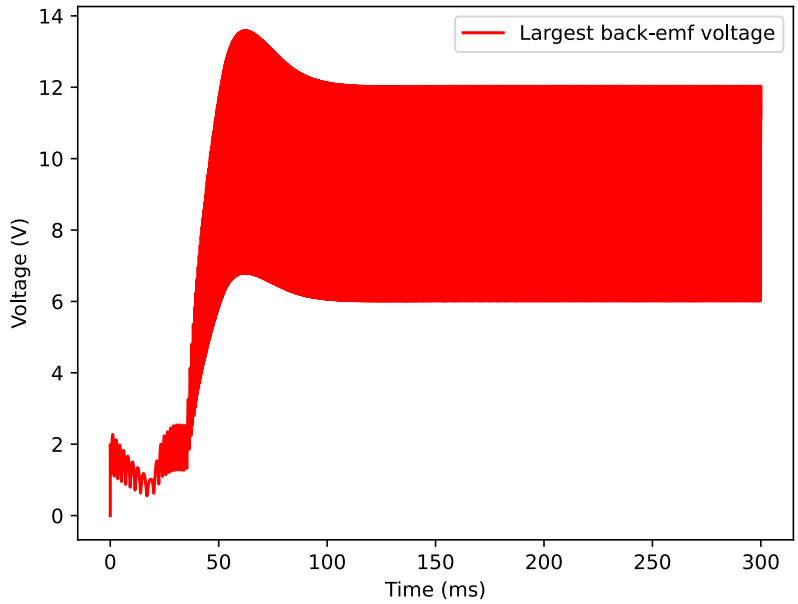

Figure G.6: The largest value for back-emf generated by a motor phase

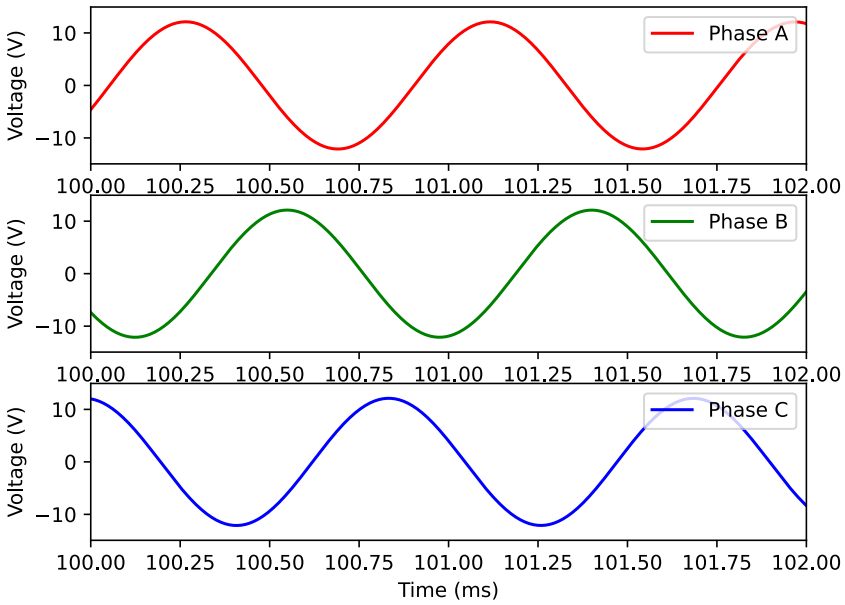

Figure G.7: Zoomed in back-emf generated by each phase

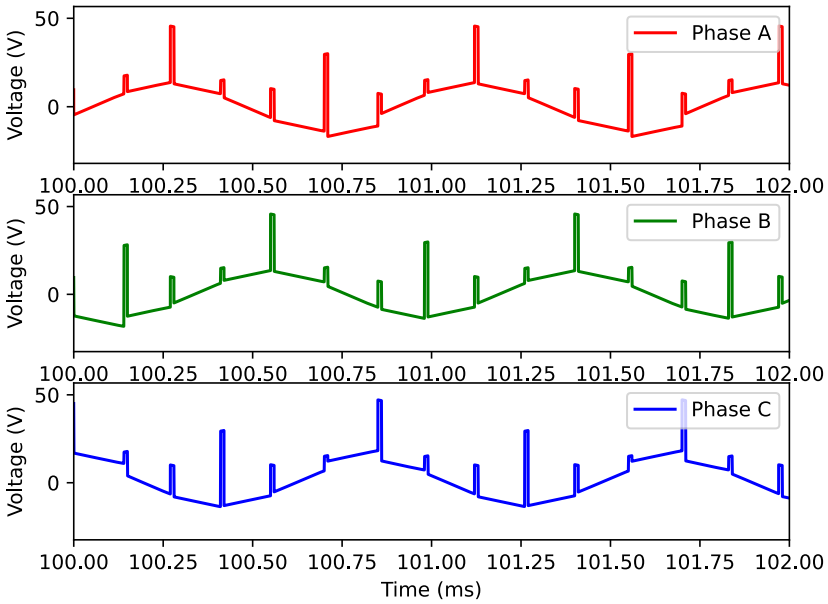

Figure G.8: Zoomed in phase voltages

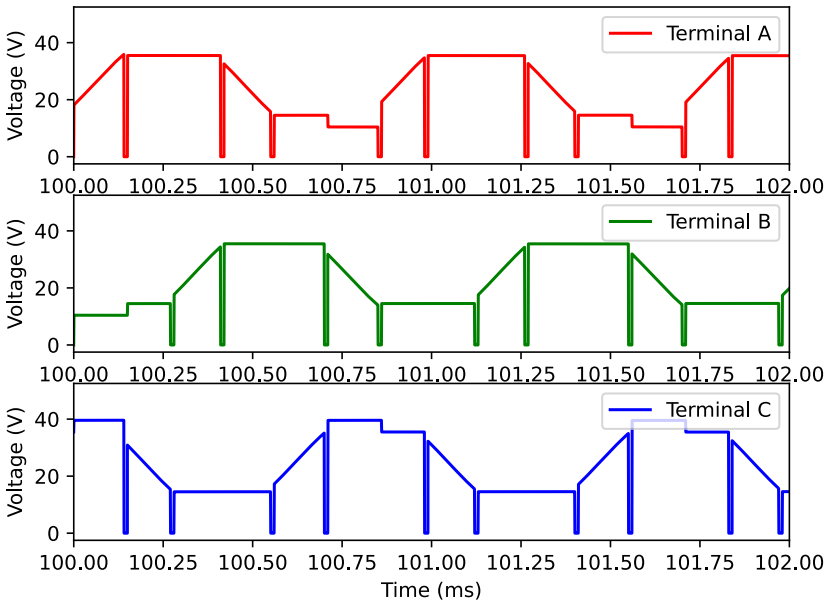

Figure G.9: Zoomed in terminal voltages

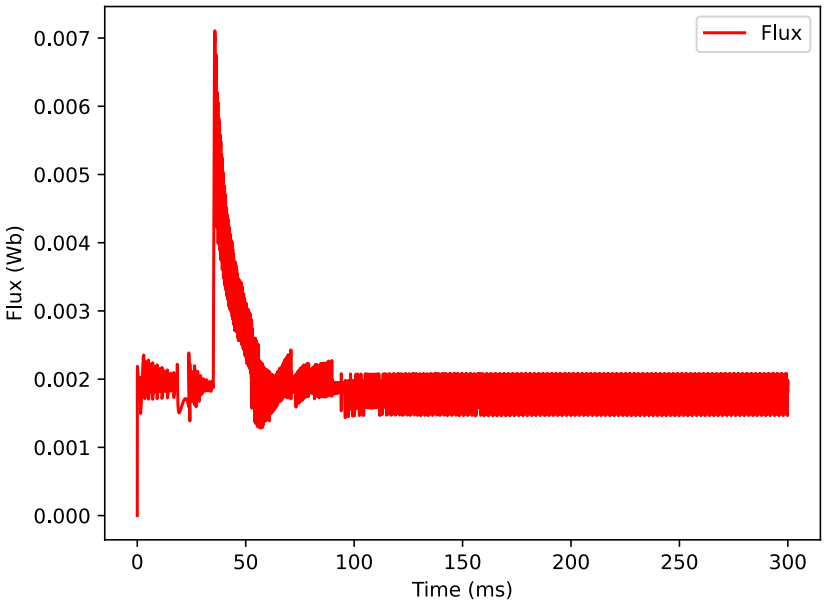

Figure G.10: Flux generated by the motor, for more information see code

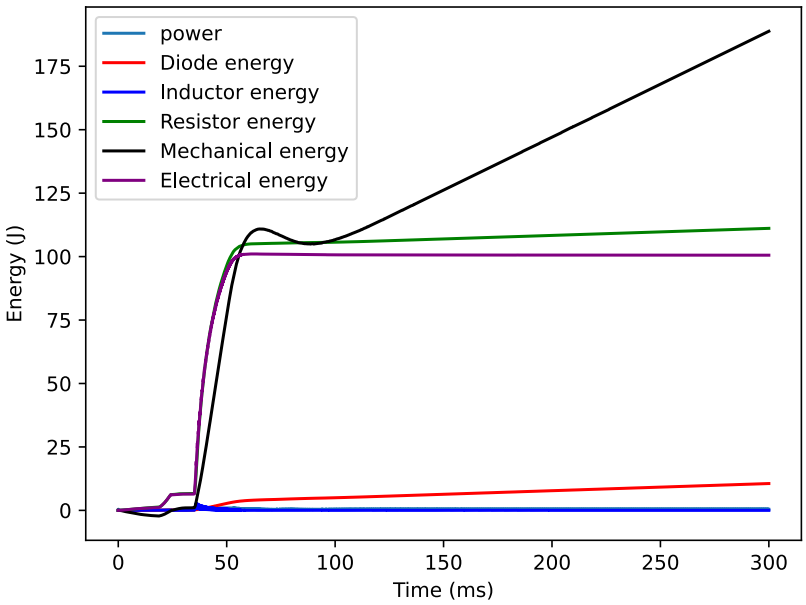

Figure G.11: 651

Appendix H

Modified trapezoidal modulation simulations

As no current limiter was implemented, the peak currents are very high, be wary about this when reading these plots. Also, the PI controller has not been fully tuned.

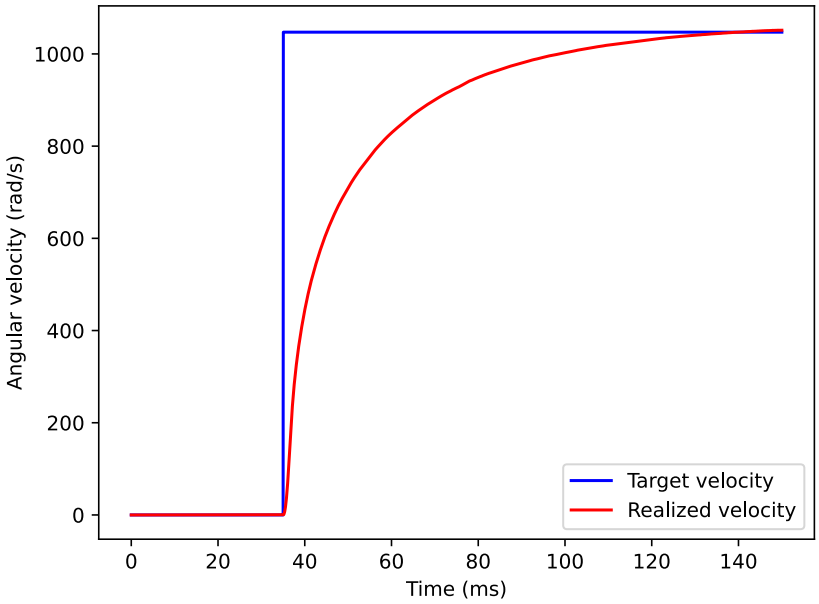

Figure H.1: The desired and realized velocity

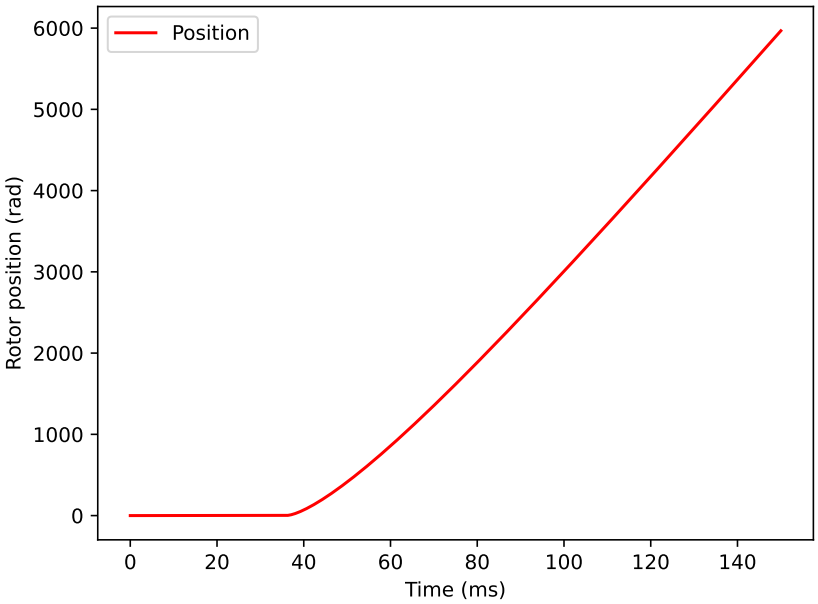

Figure H.2: The position of the rotor

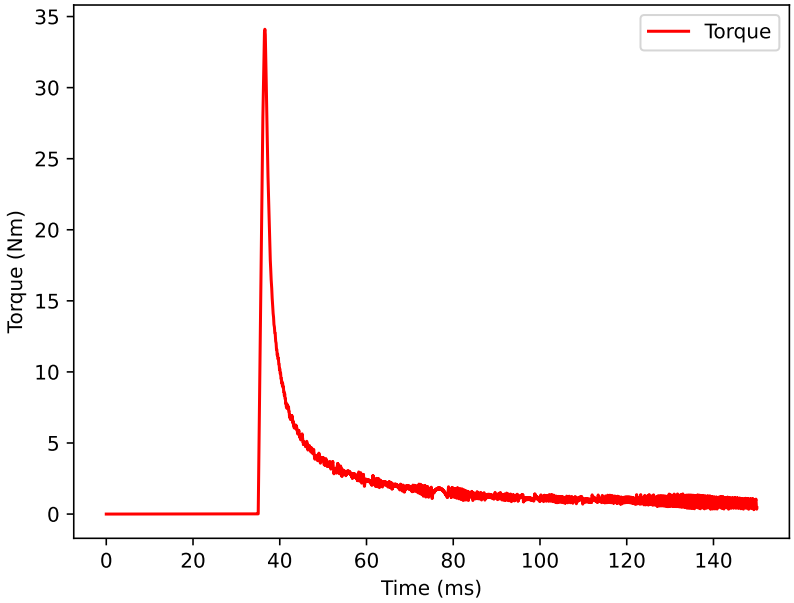

Figure H.3: The torque delivered by the motor

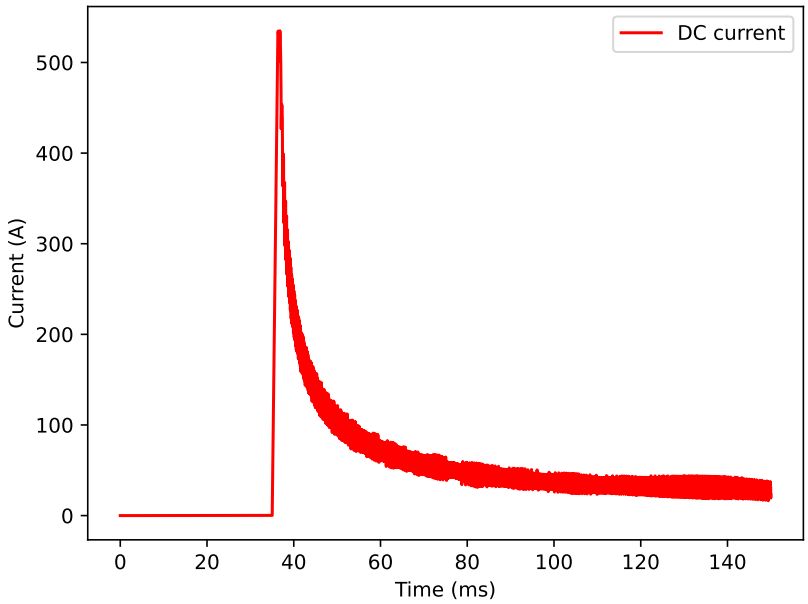

Figure H.4: The DC current through the motor

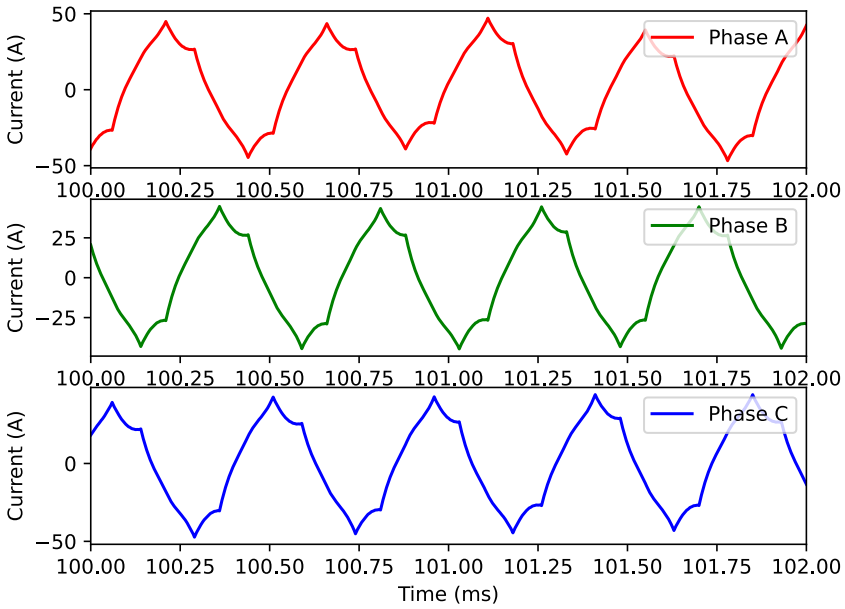

Figure H.5: Zoomed in phase currents through the motor

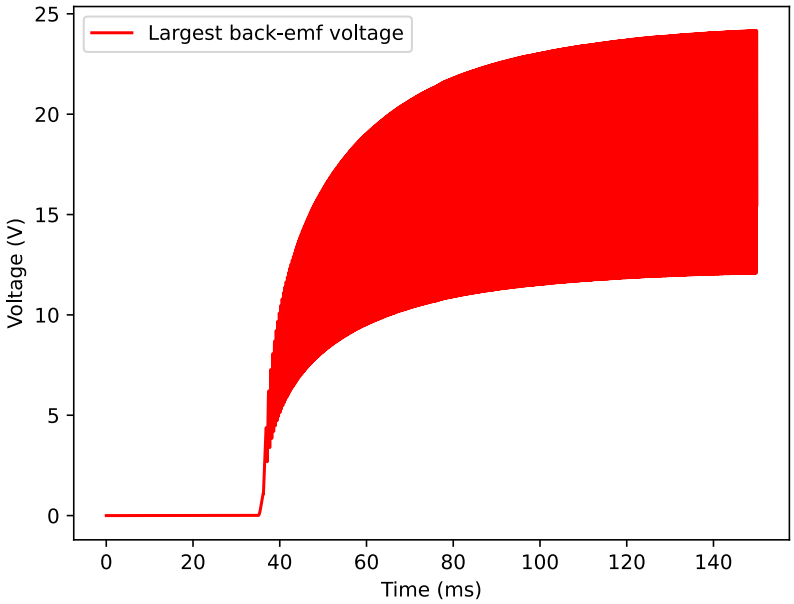

Figure H.6: The largest value for back-emf generated by a motor phase

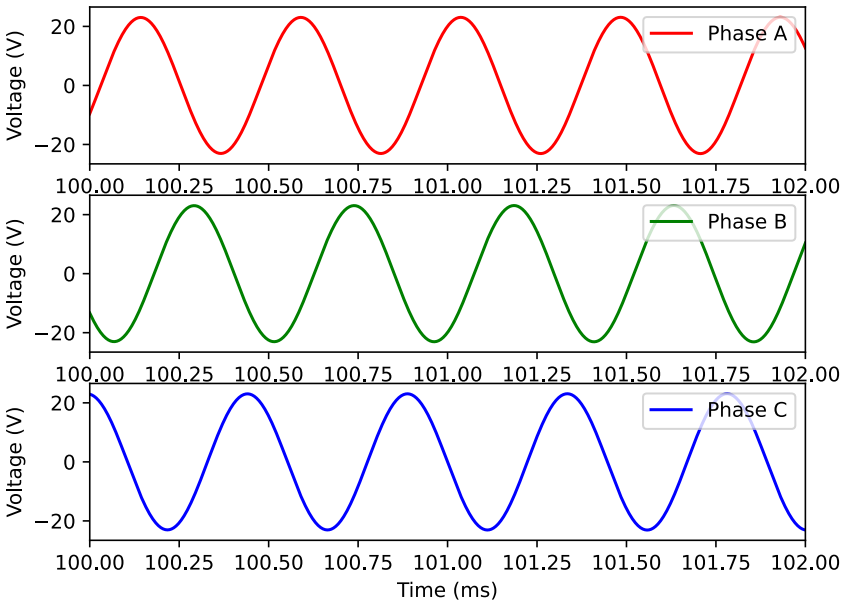

Figure H.7: Zoomed in back-emf generated by each phase

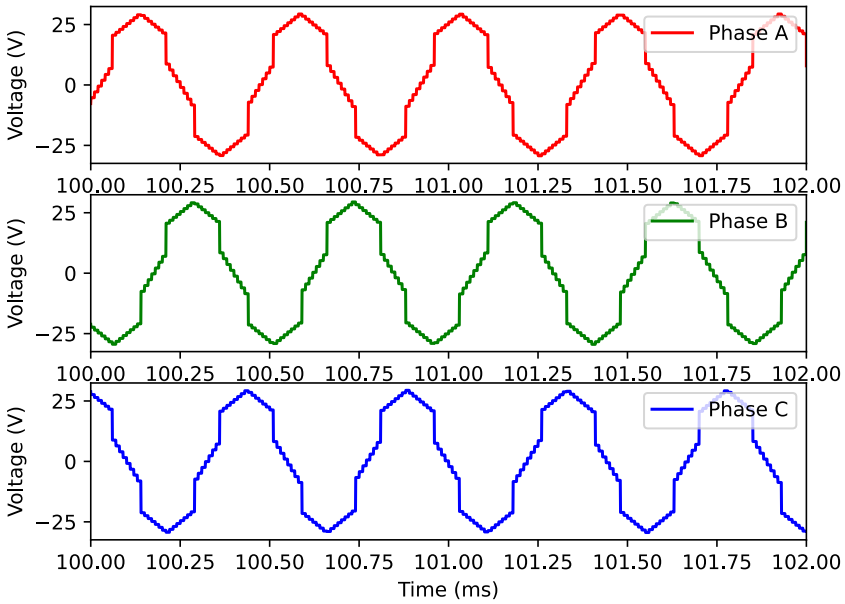

Figure H.8: Zoomed in phase voltages

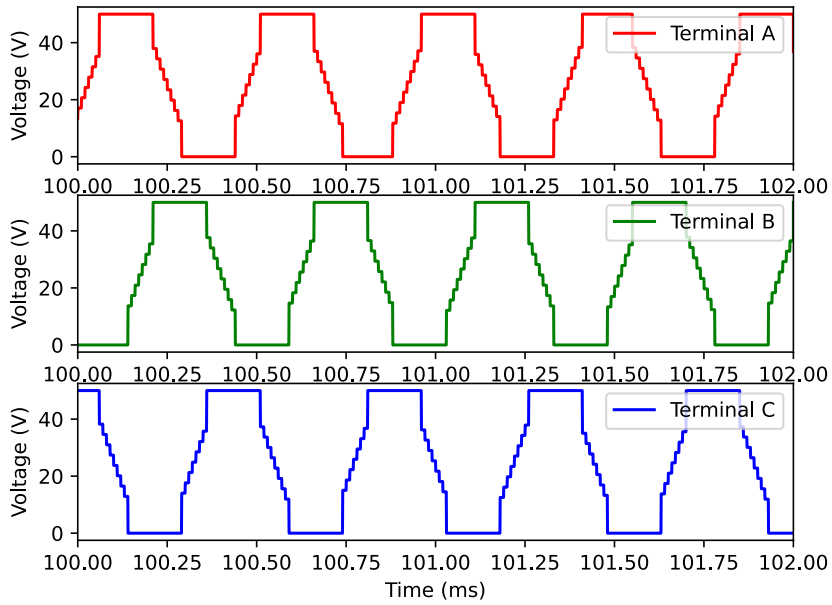

Figure H.9: Zoomed in terminal voltages

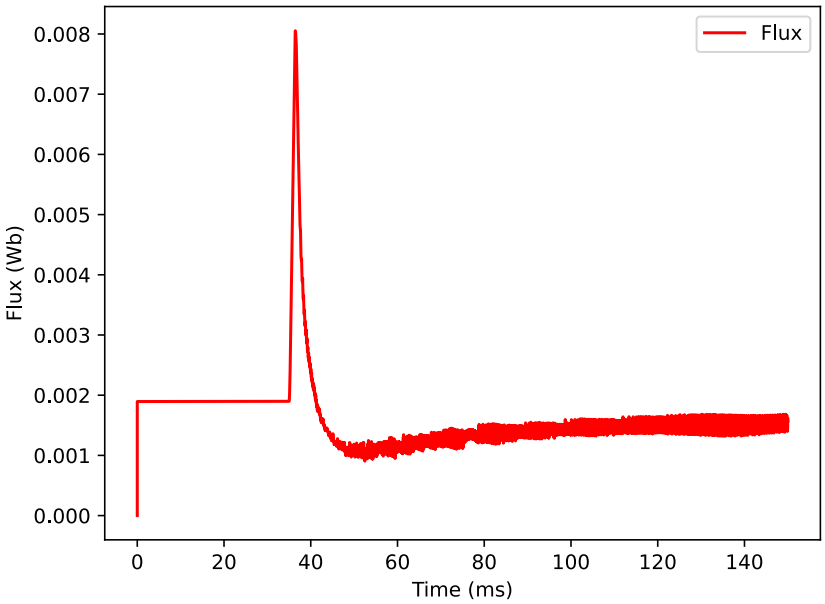

Figure H.10: Flux generated by the motor, for more information see code

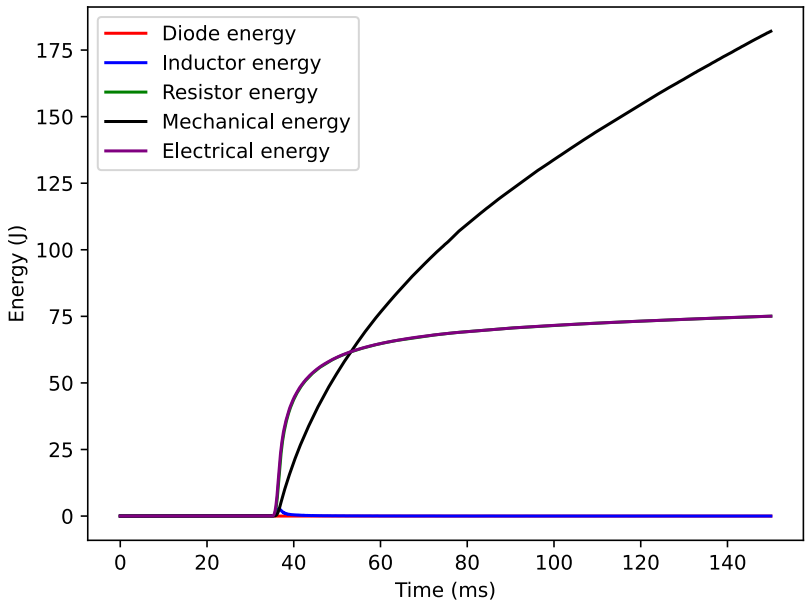

Figure H.11: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.708

Appendix I

Benchmark simulation results

Maximum velocity with a small propeller

Trapezoidal control

Required velocity

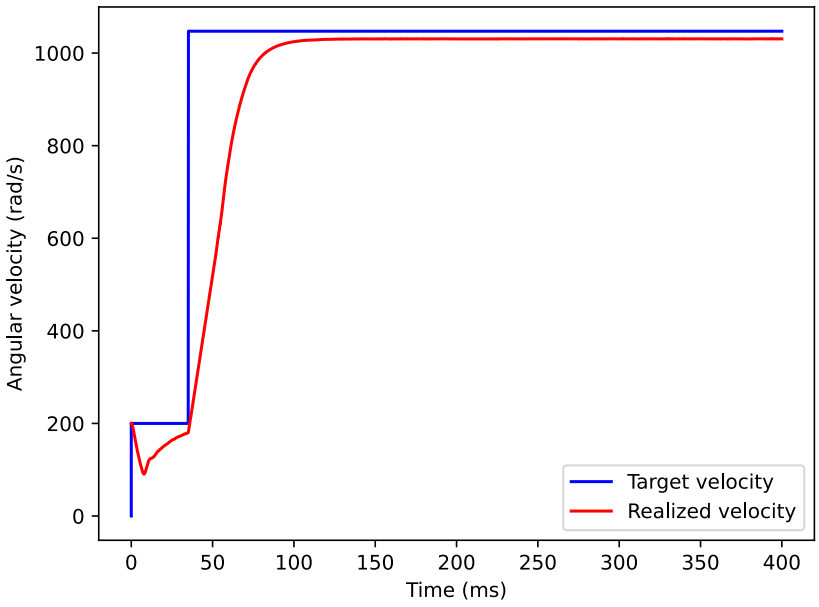

Figure I.1: The desired and realized velocity

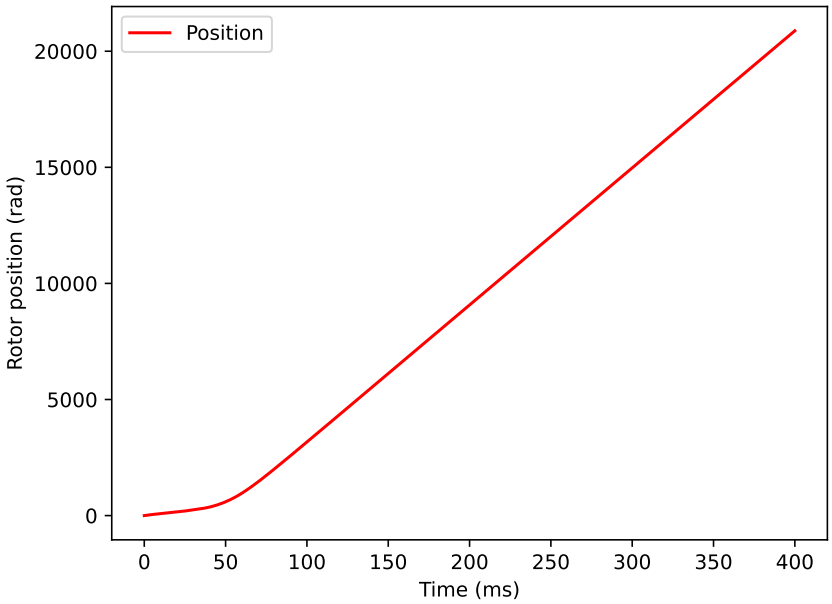

Figure I.2: The position of the rotor

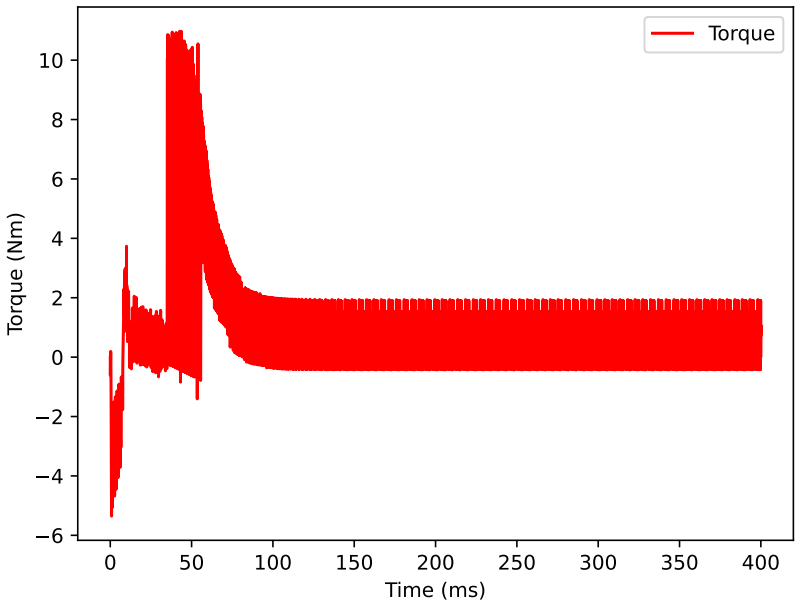

Figure I.3: The torque delivered by the motor

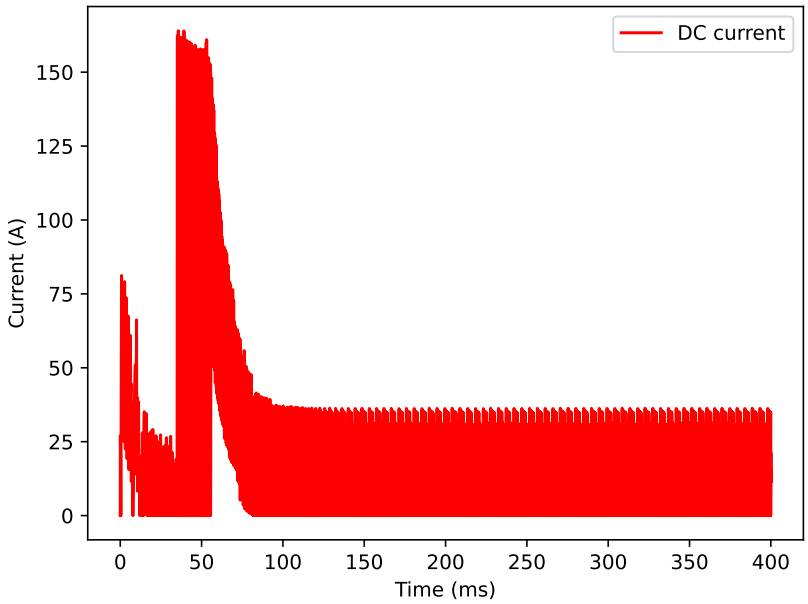

Figure I.4: The DC current through the motor

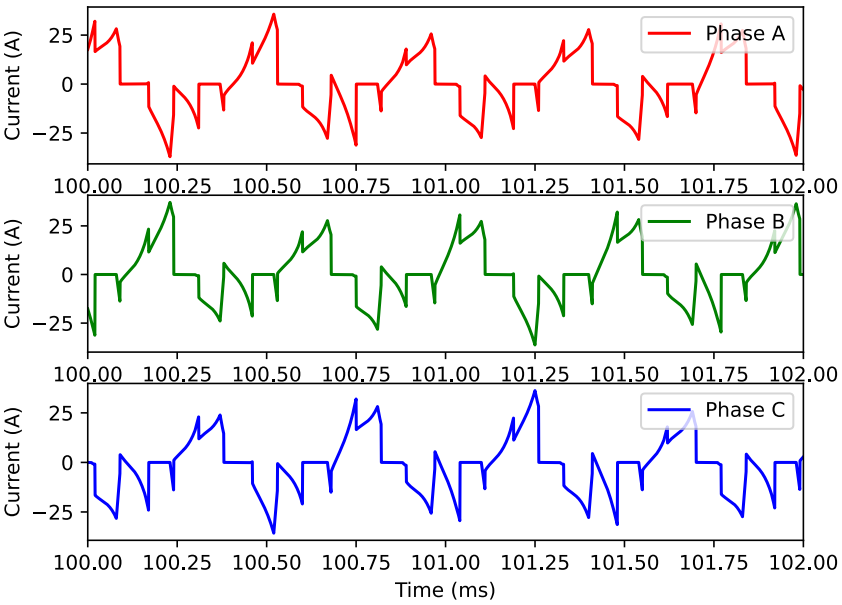

Figure I.5: Zoomed in phase currents through the motor

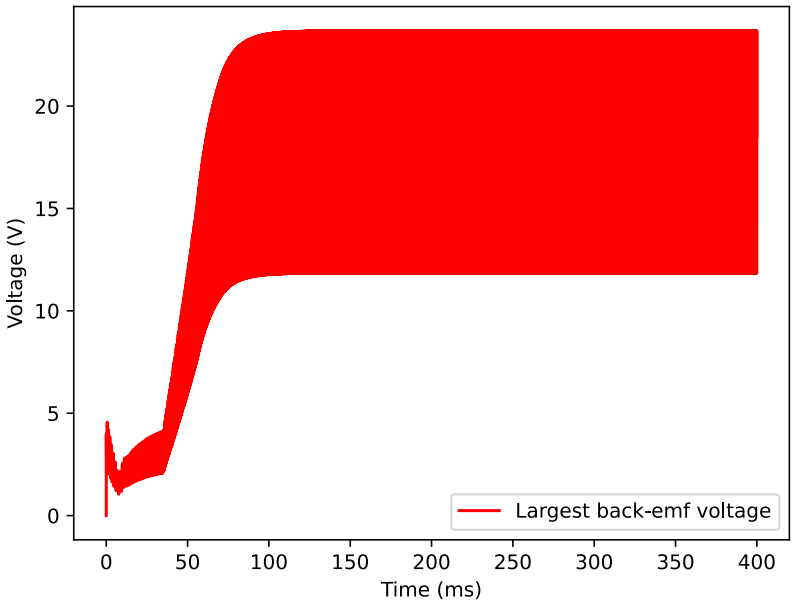

Figure I.6: The largest value for back-emf generated by a motor phase

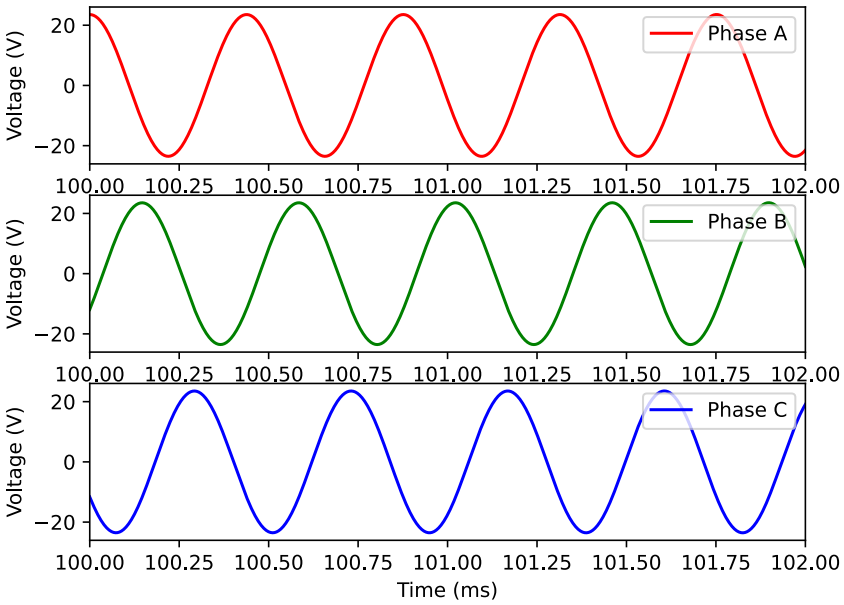

Figure I.7: Zoomed in back-emf generated by each phase

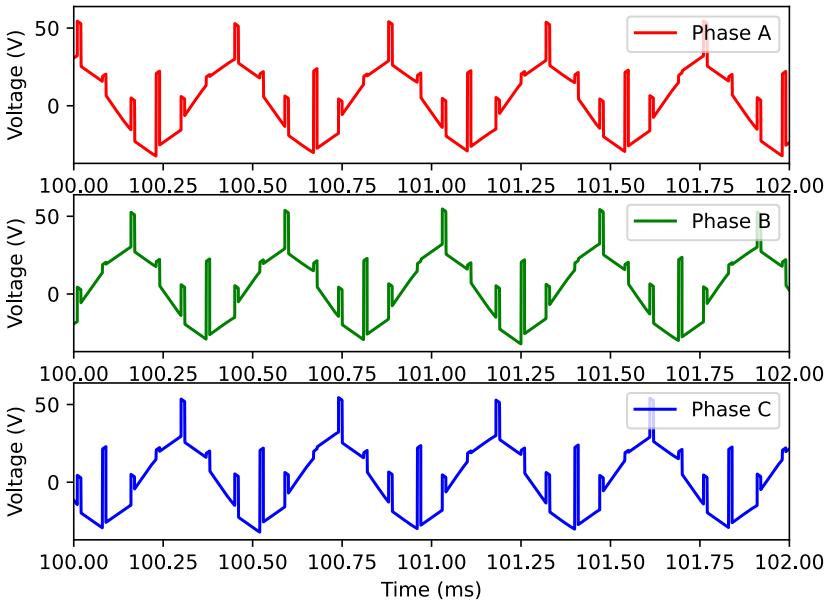

Figure I.8: Zoomed in phase voltages

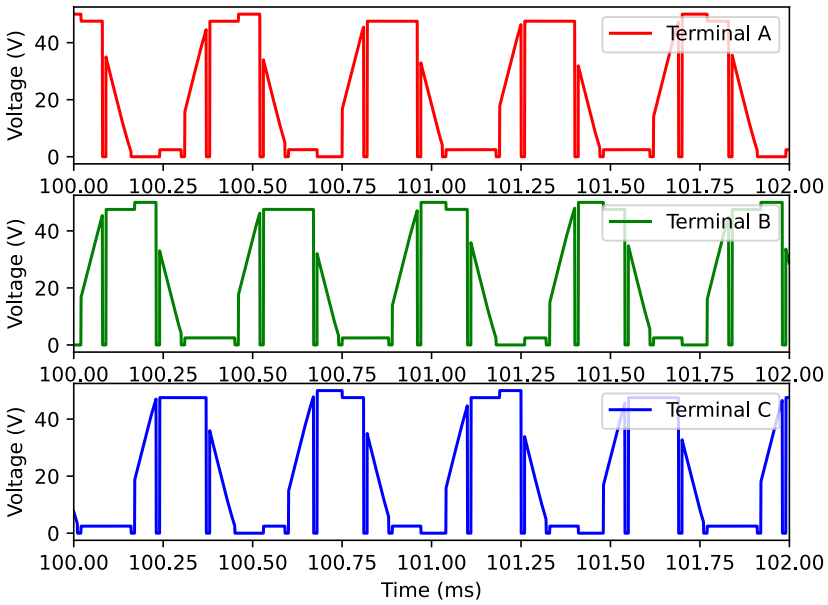

Figure I.9: Zoomed in terminal voltages

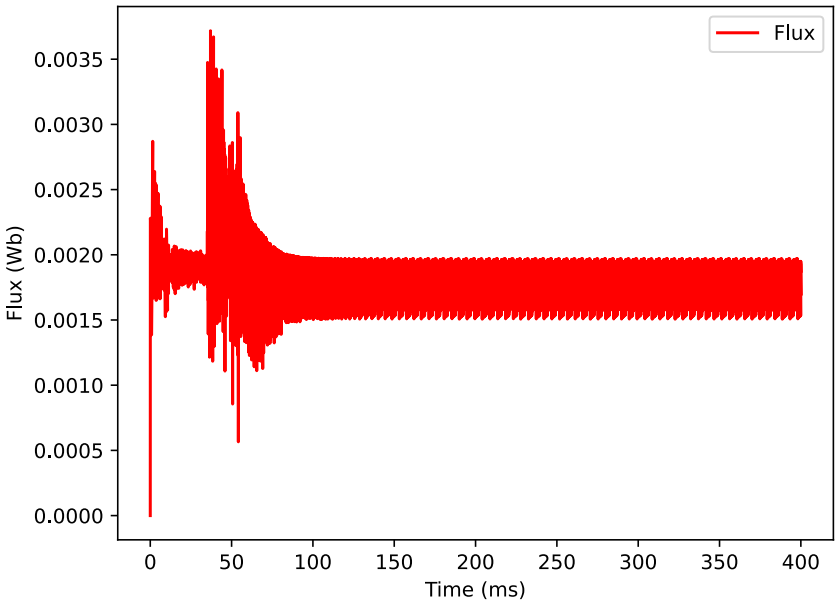

Figure I.10: Flux generated by the motor, for more information see code

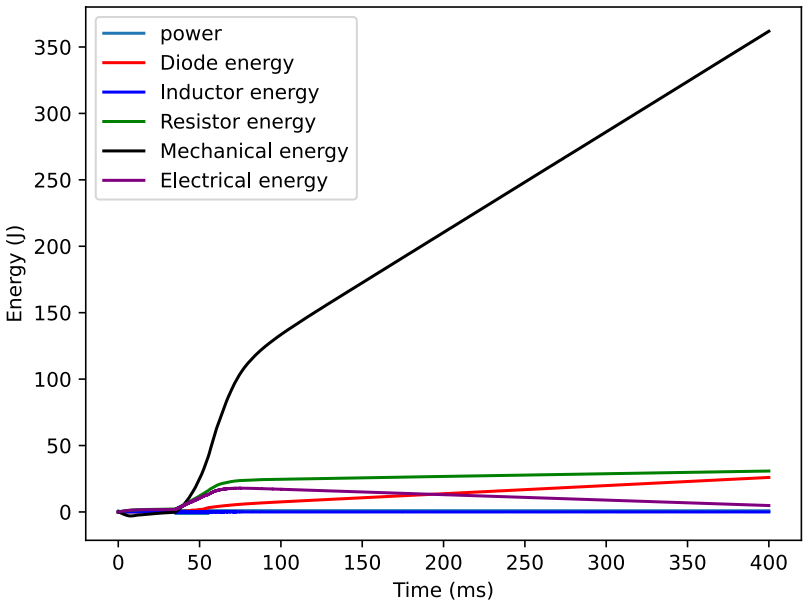

Figure I.11: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.977

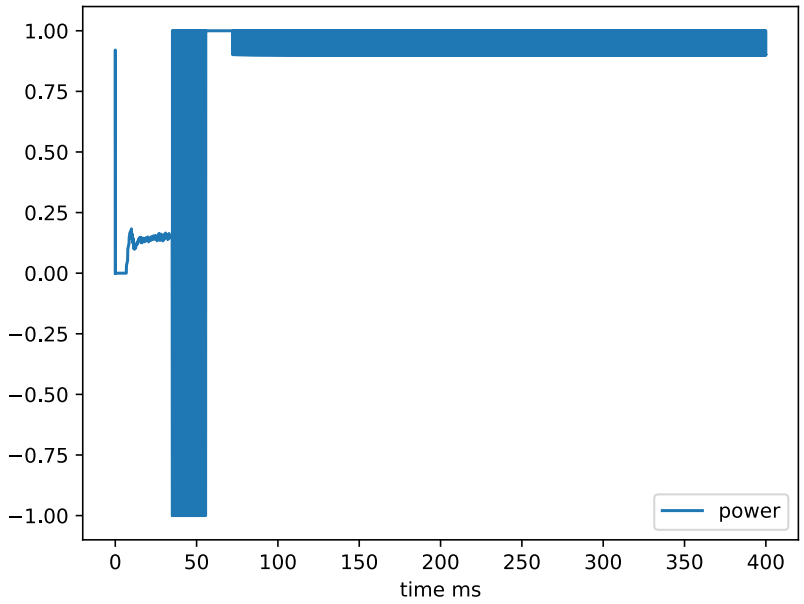

Figure I.12: The amplitude of the normalized output voltage

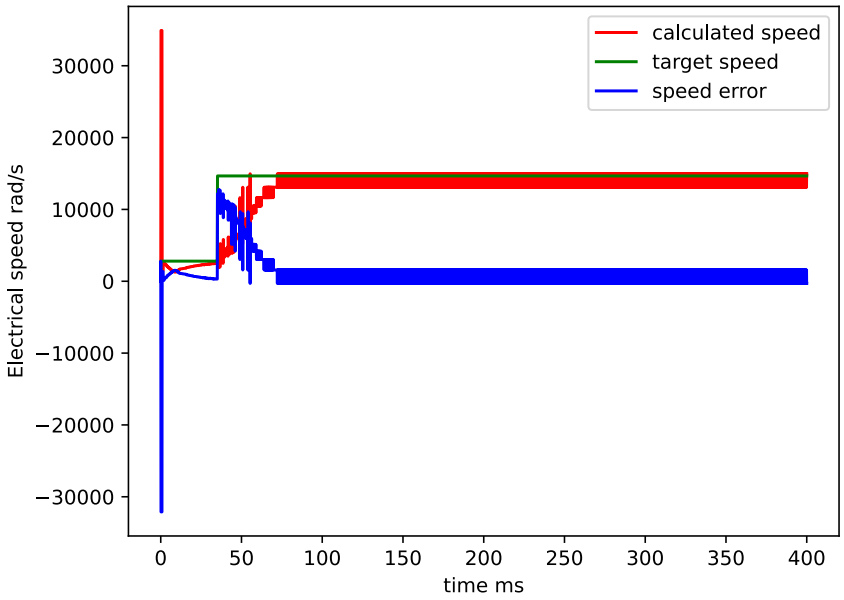

Figure I.13: The calculated, target, and error in the electrical speed of the trapezoidal controller

Double required velocity

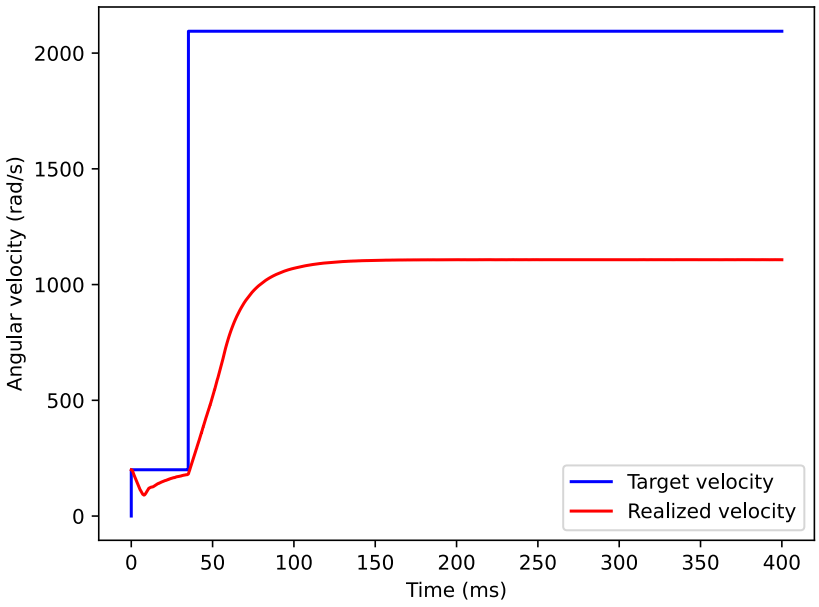

Figure I.14: The desired and realized velocity

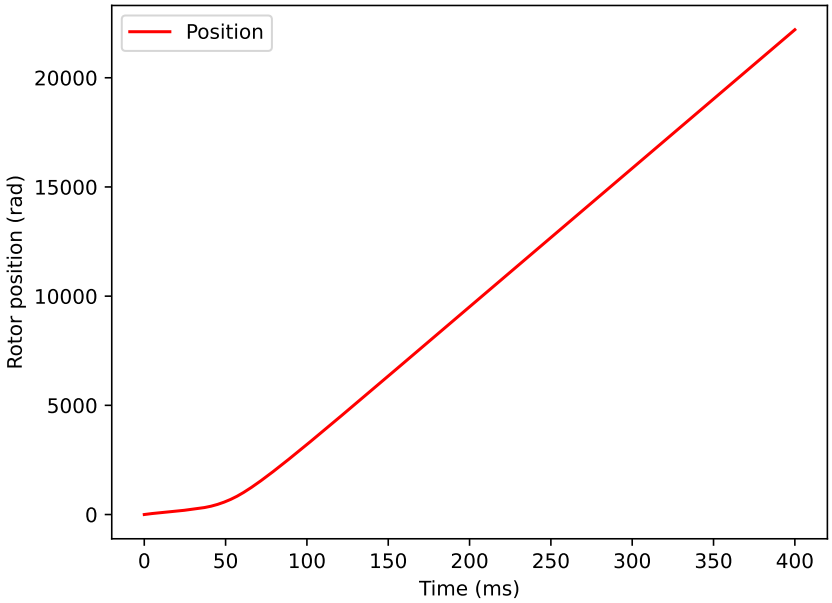

Figure I.15: The position of the rotor

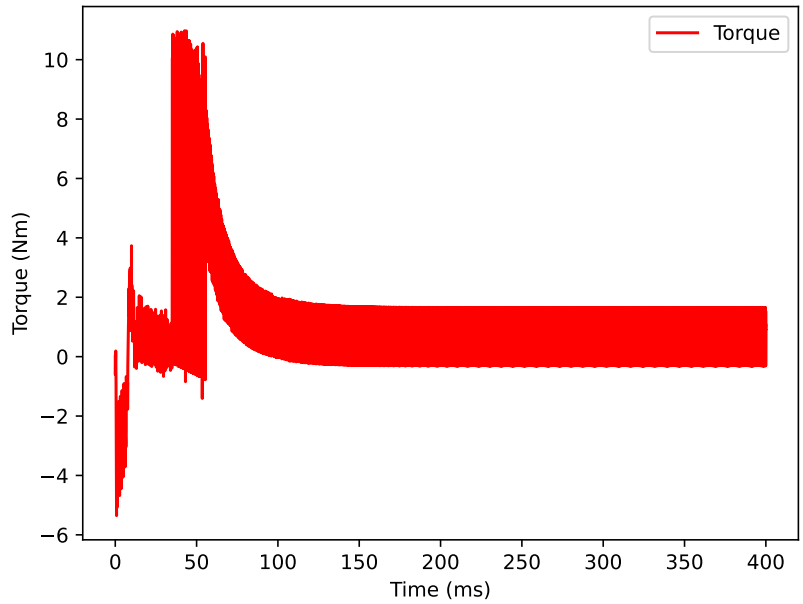

Figure I.16: The torque delivered by the motor

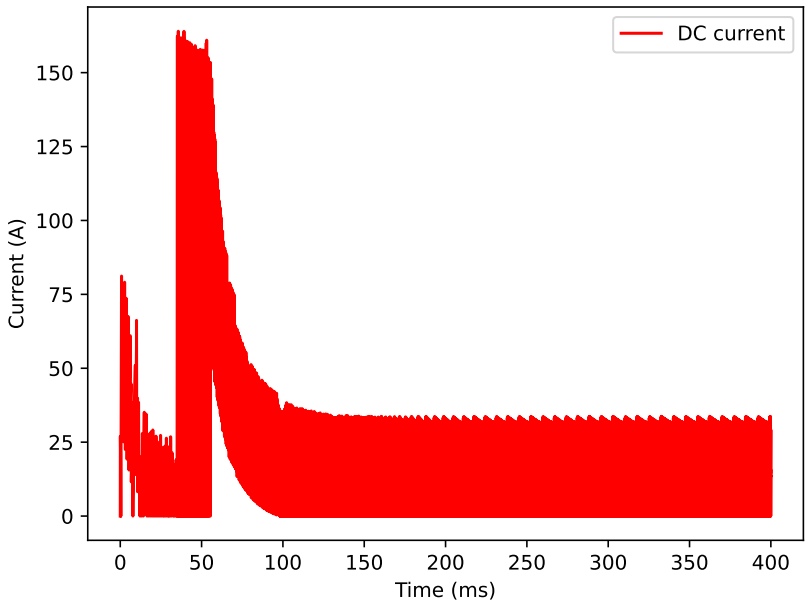

Figure I.17: The DC current through the motor

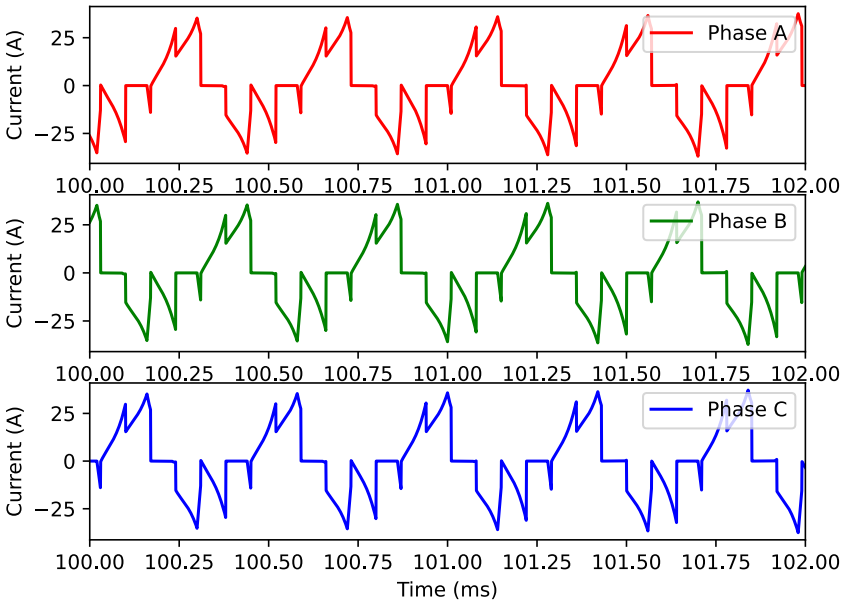

Figure I.18: Zoomed in phase currents through the motor

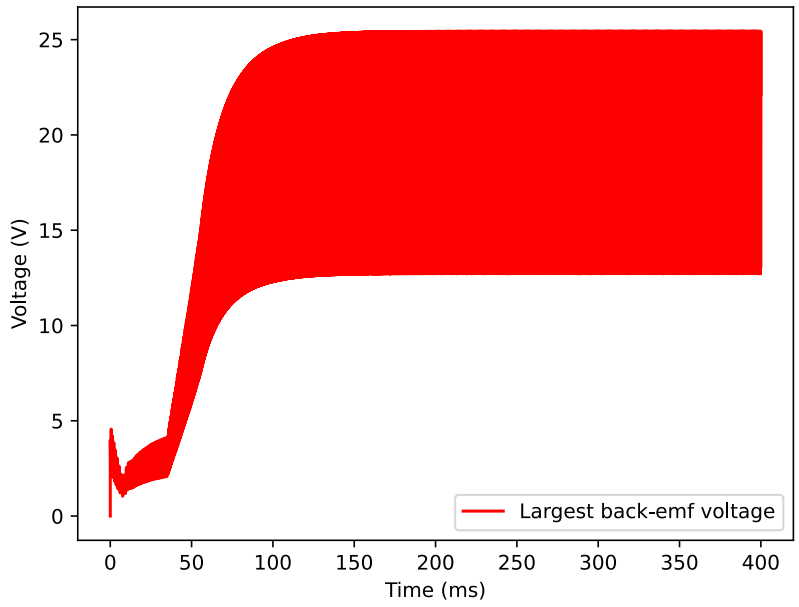

Figure I.19: The largest value for back-emf generated by a motor phase

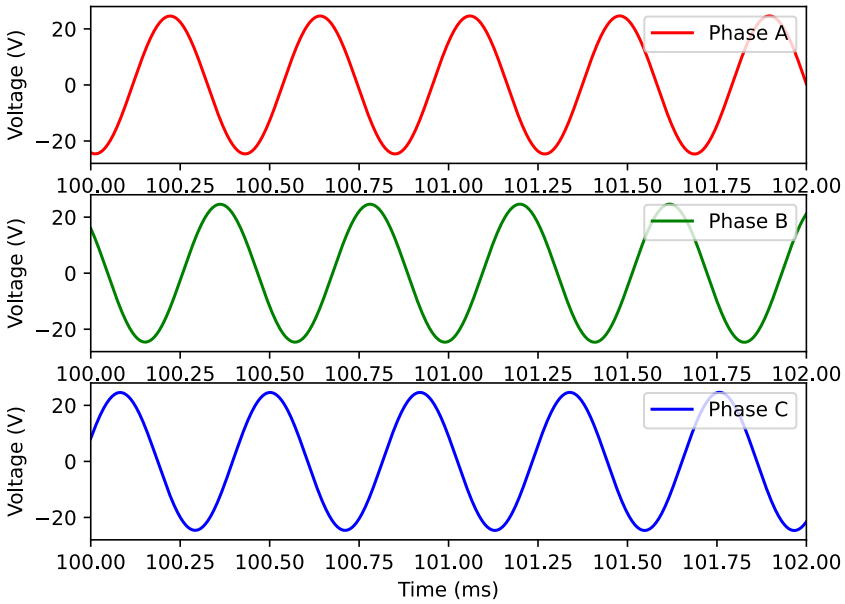

Figure I.20: Zoomed in back-emf generated by each phase

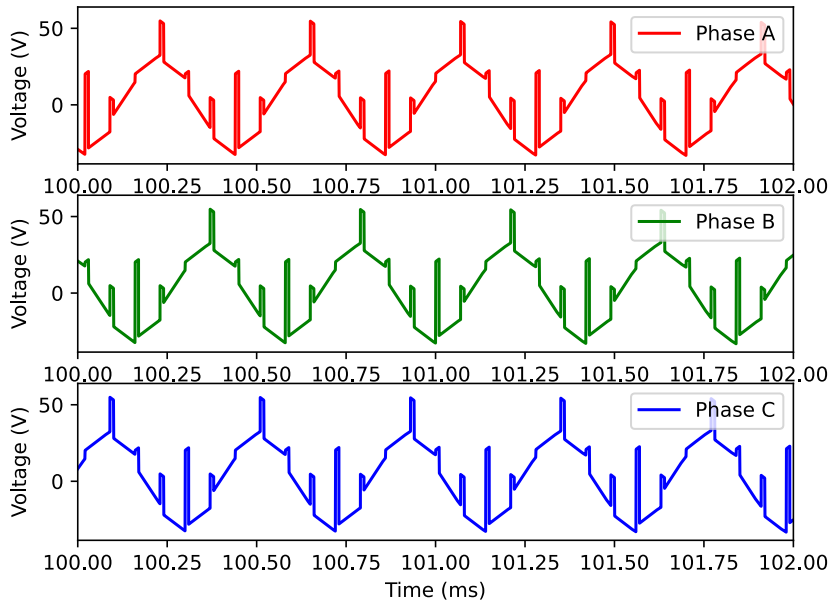

Figure I.21: Zoomed in phase voltages

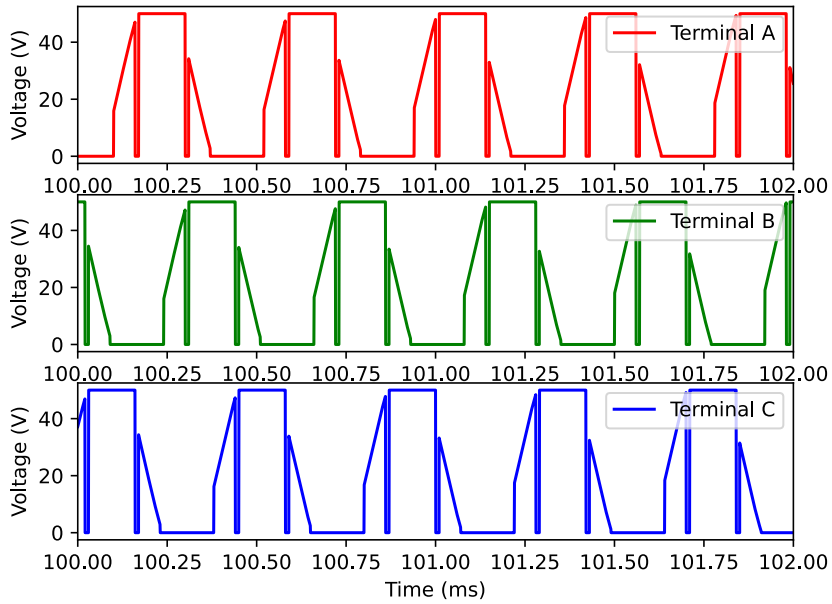

Figure I.22: Zoomed in terminal voltages

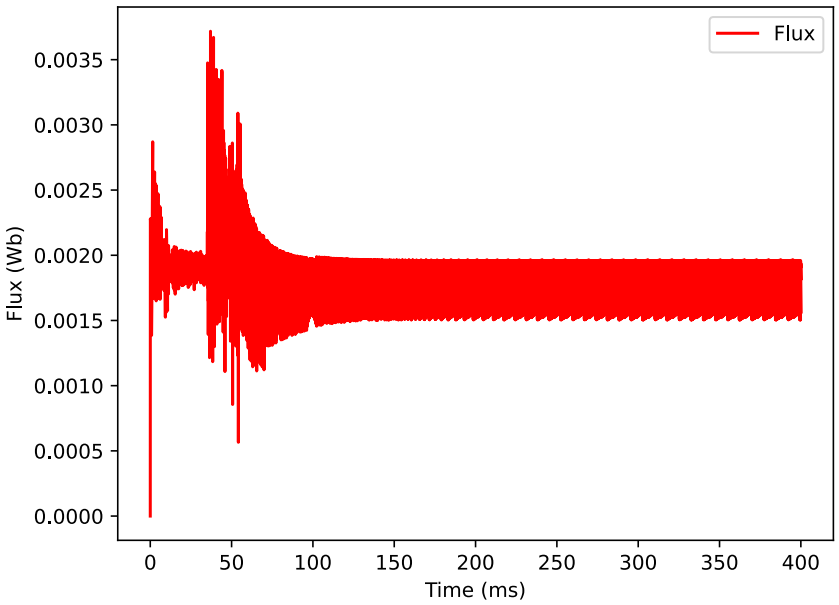

Figure I.23: Flux generated by the motor, for more information see code

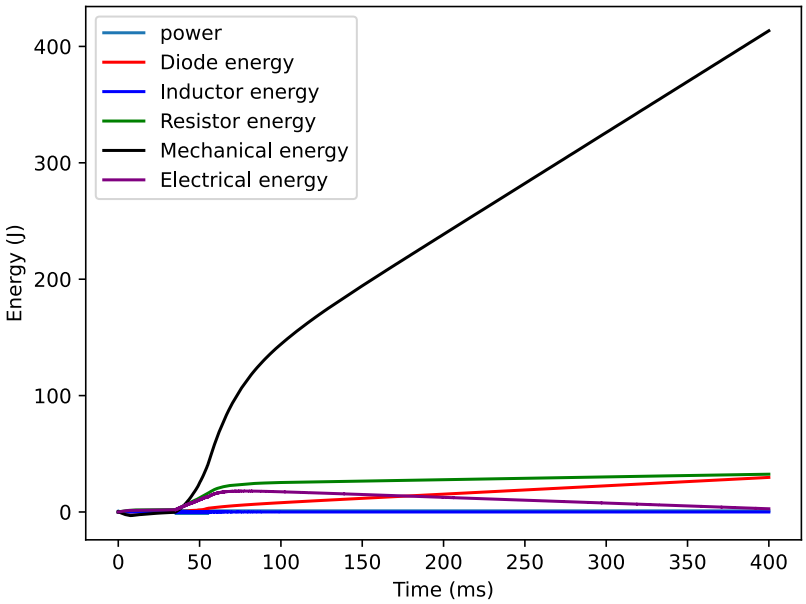

Figure I.24: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.993

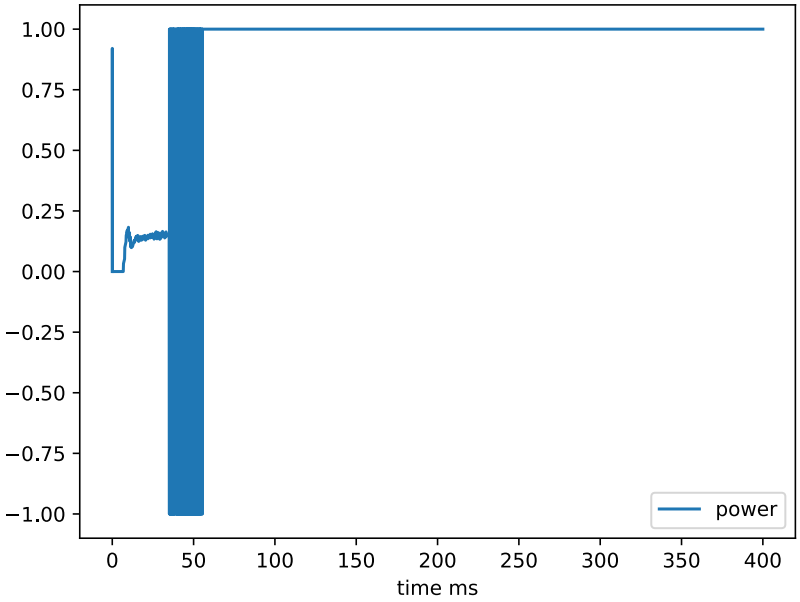

Figure I.25: The amplitude of the normalized output voltage

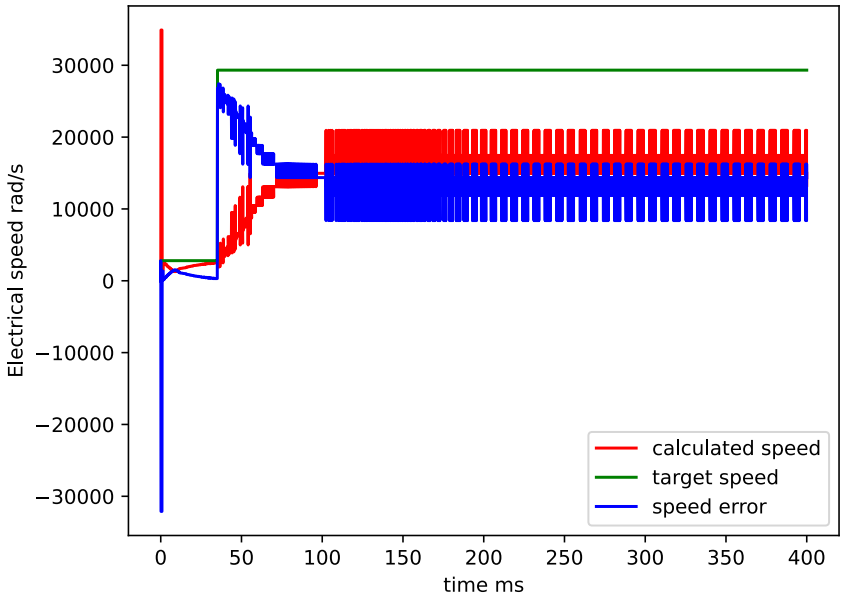

Figure I.26: The calculated, target, and error in the electrical speed of the trapezoidal controller

FOC

Required velocity

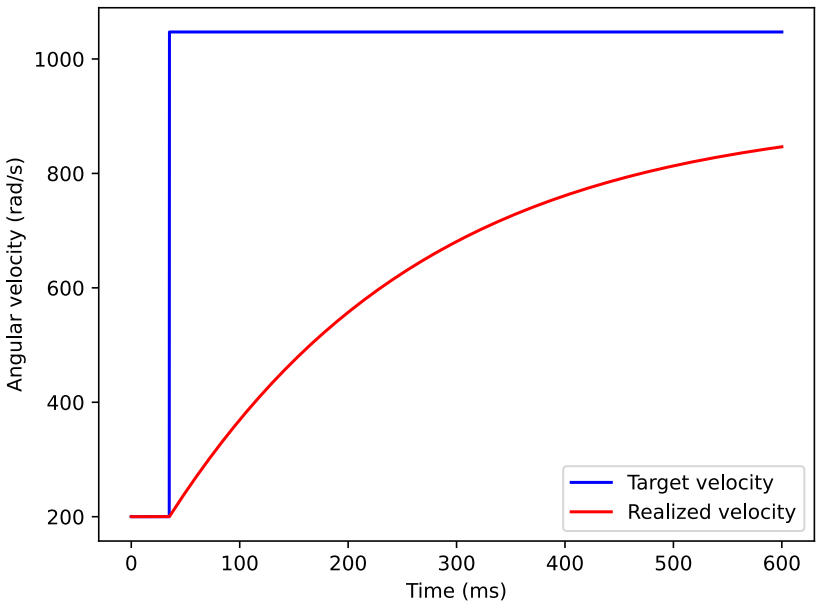

Figure I.27: The desired and realized velocity

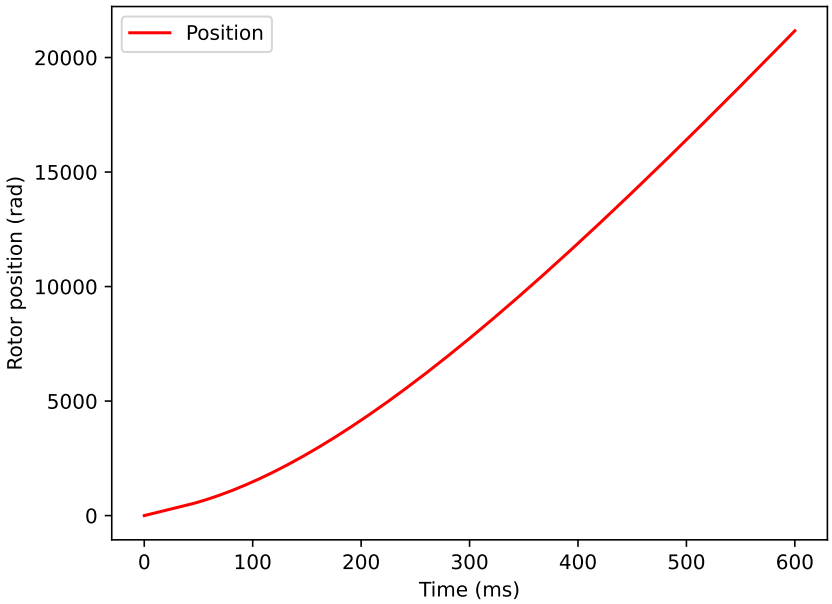

Figure I.28: The position of the rotor

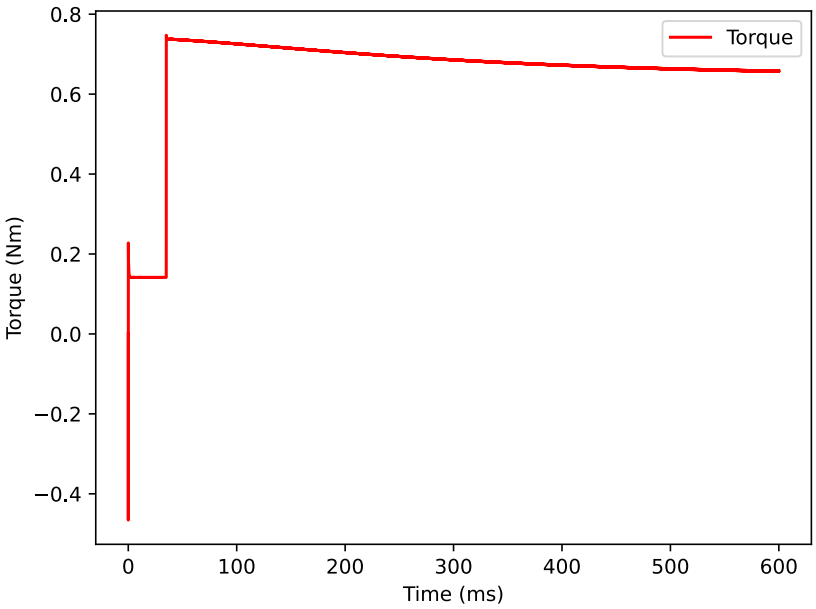

Figure I.29: The torque delivered by the motor

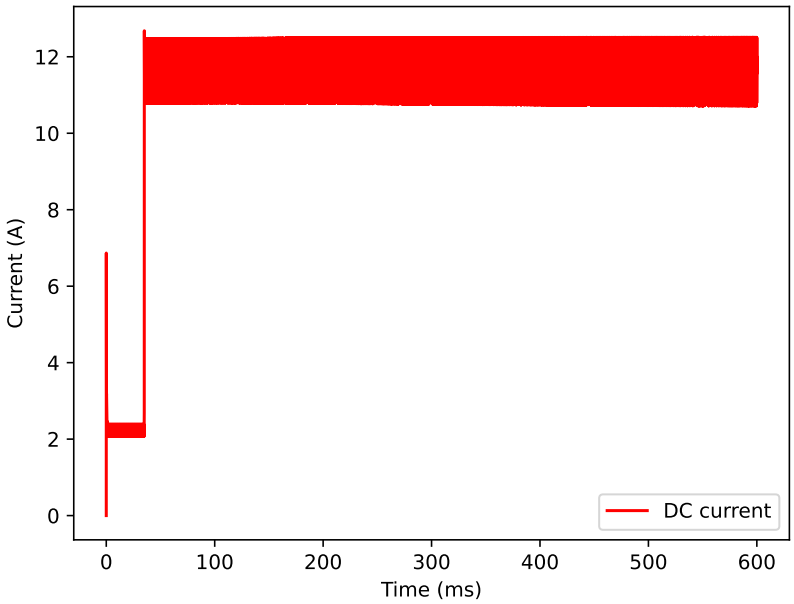

Figure I.30: The DC current through the motor

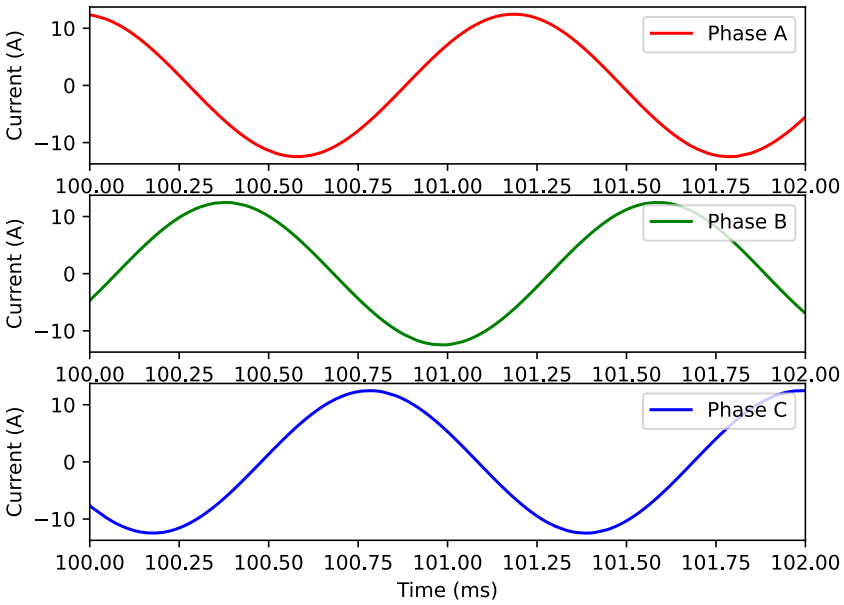

Figure I.31: Zoomed in phase currents through the motor

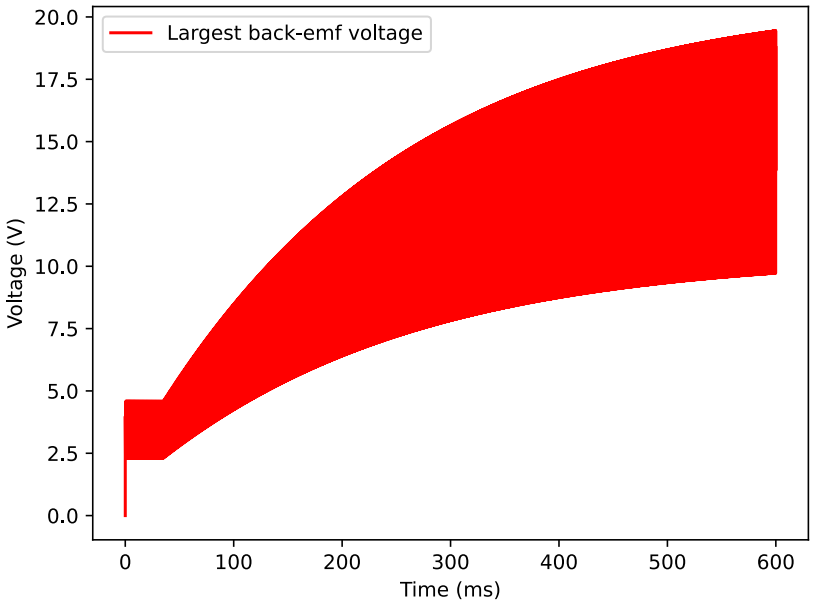

Figure I.32: The largest value for back-emf generated by a motor phase

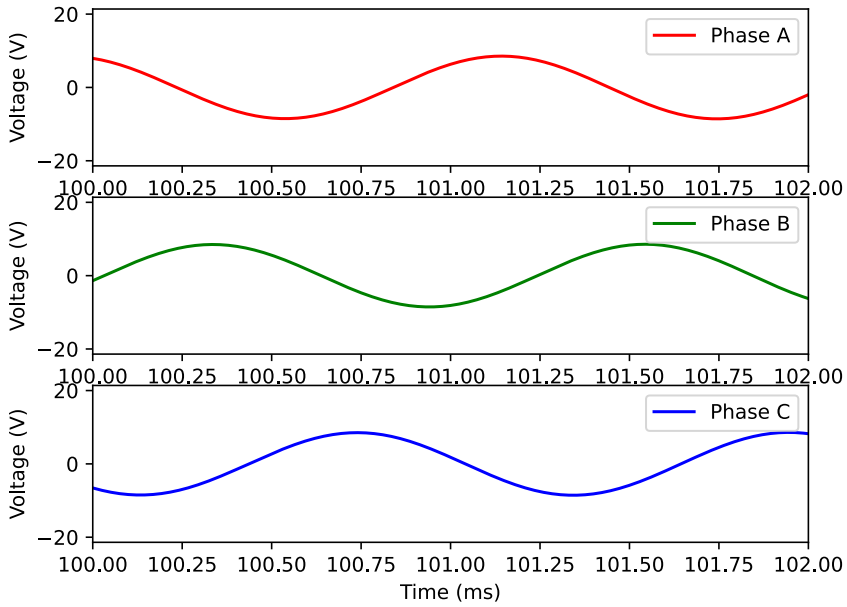

Figure I.33: Zoomed in back-emf generated by each phase

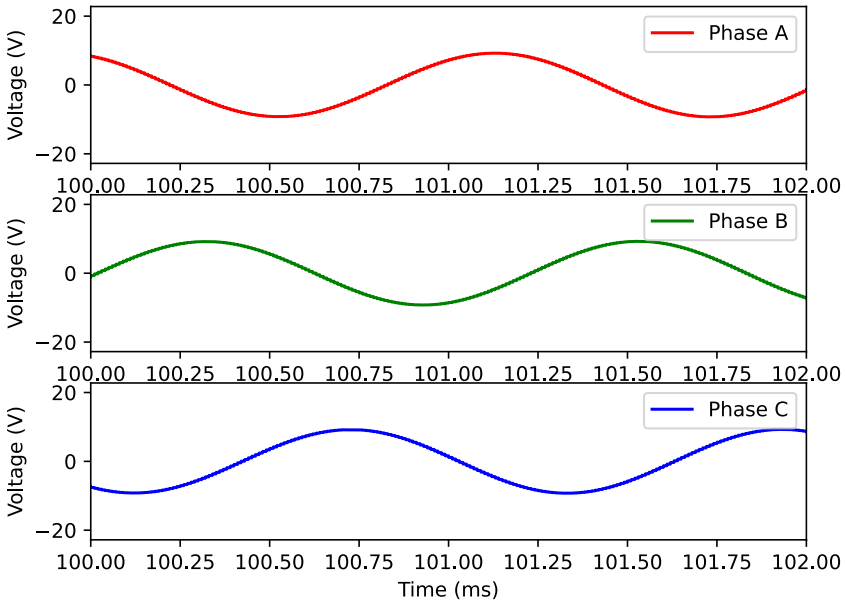

Figure I.34: Zoomed in phase voltages

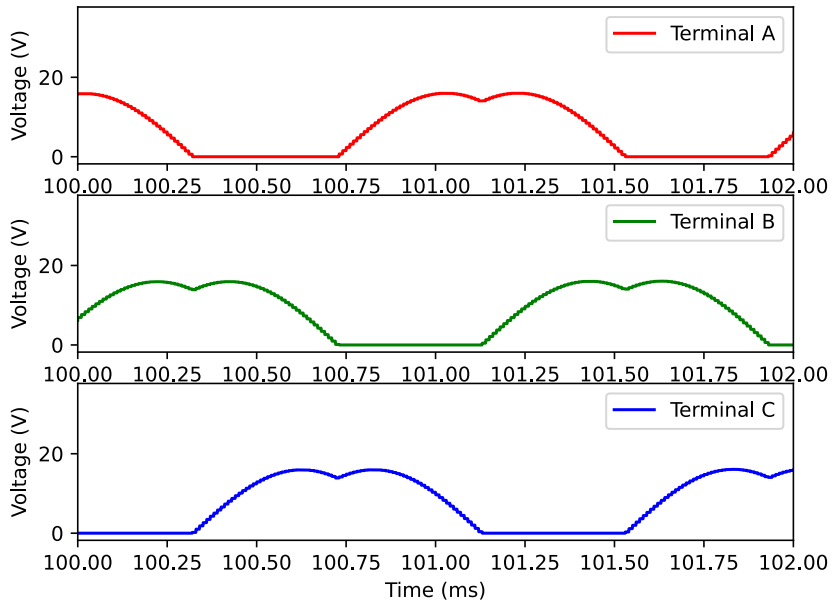

Figure I.35: Zoomed in terminal voltages

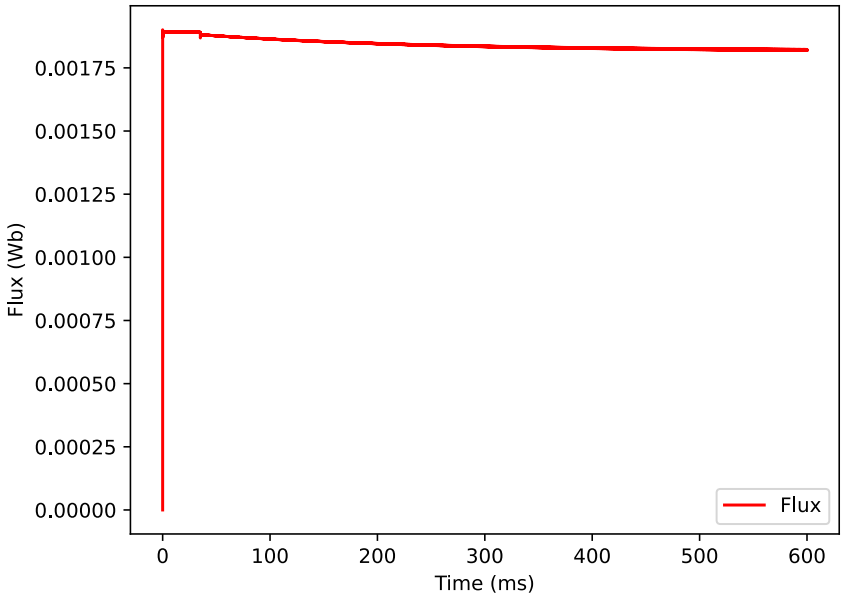

Figure I.36: Flux generated by the motor, for more information see code

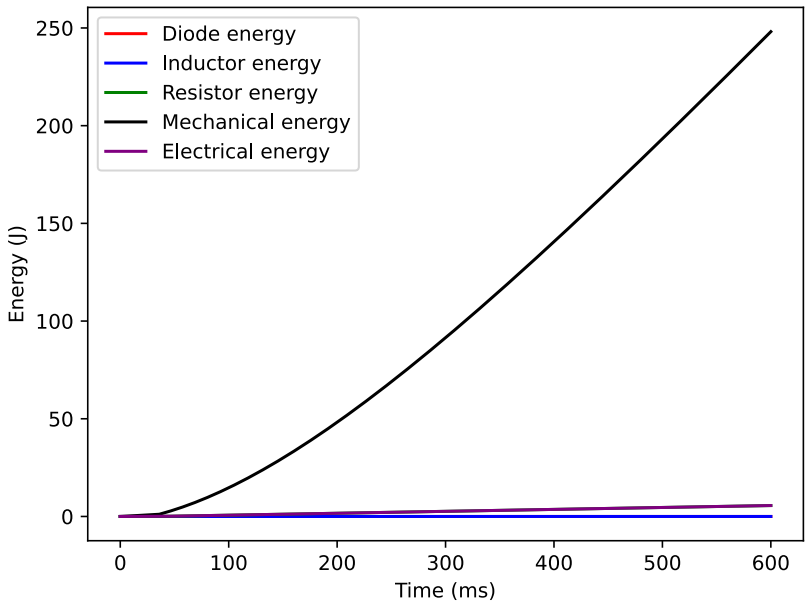

Figure I.37: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.978

Double required velocity

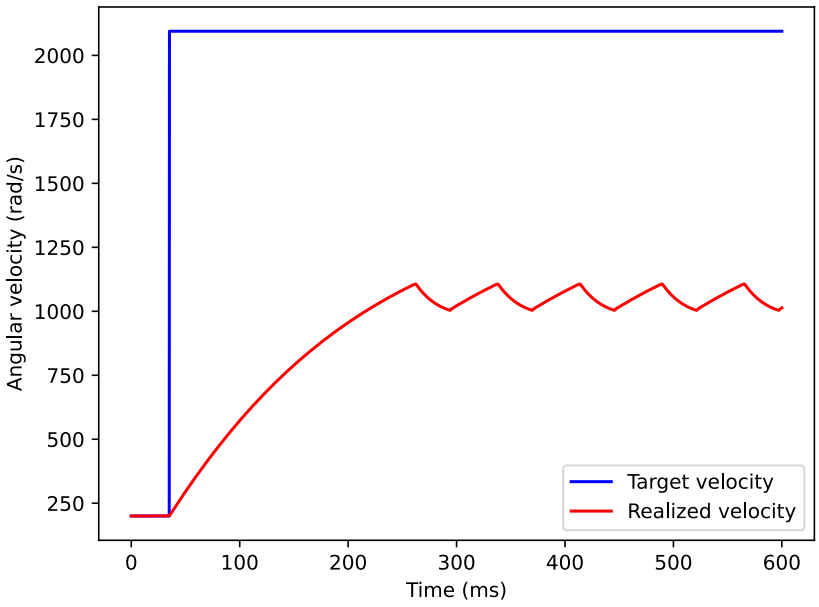

Figure I.38: The desired and realized velocity

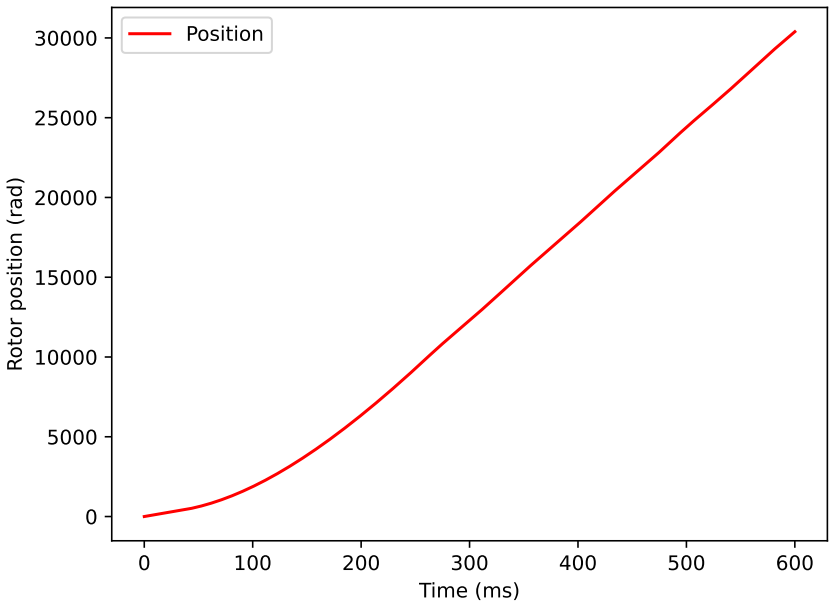

Figure I.39: The position of the rotor

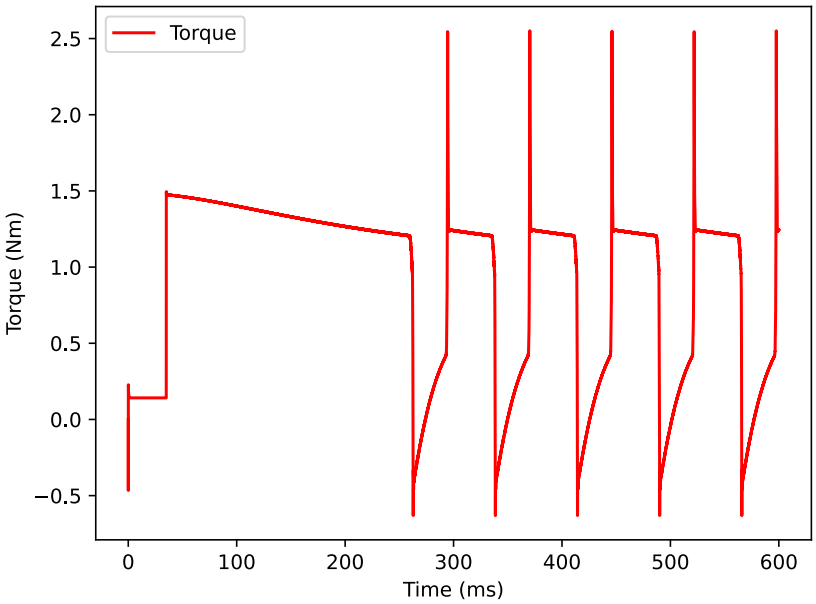

Figure I.40: The torque delivered by the motor

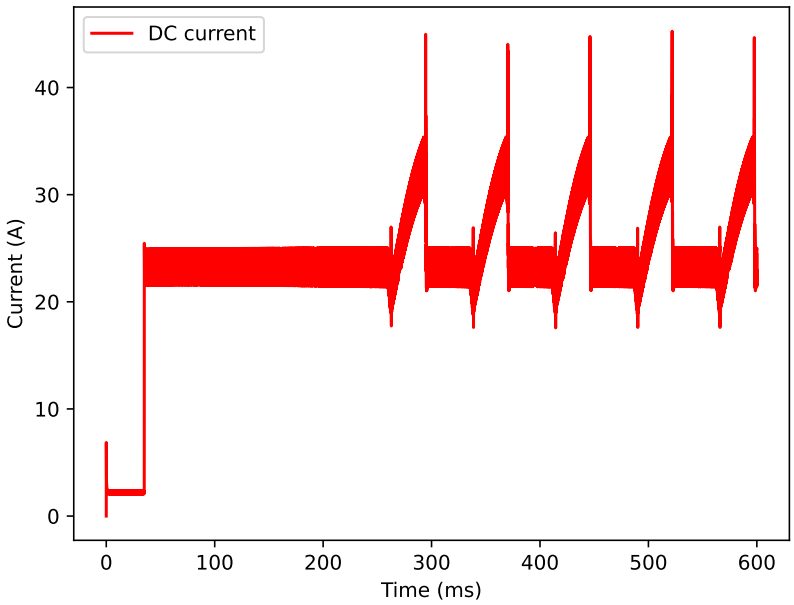

Figure I.41: The DC current through the motor

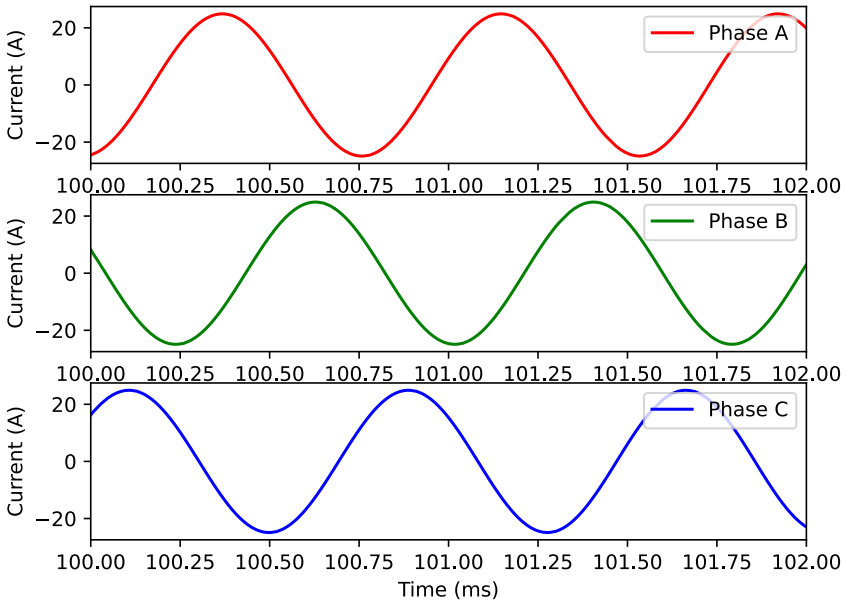

Figure I.42: Zoomed in phase currents through the motor

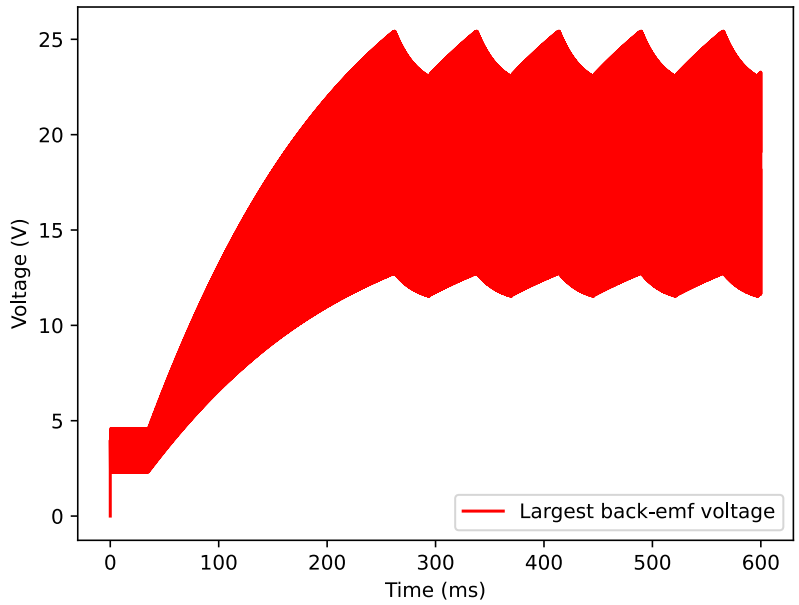

Figure I.43: The largest value for back-emf generated by a motor phase

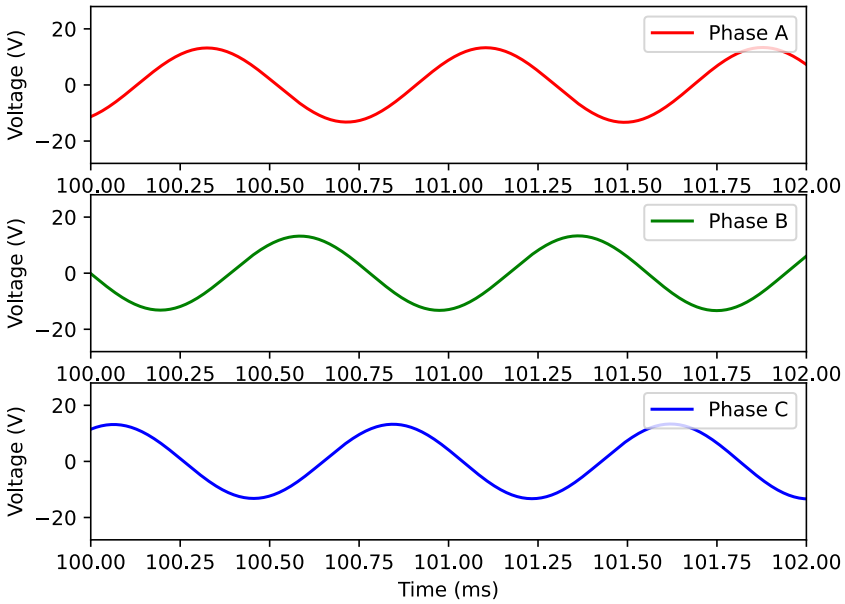

Figure I.44: Zoomed in back-emf generated by each phase

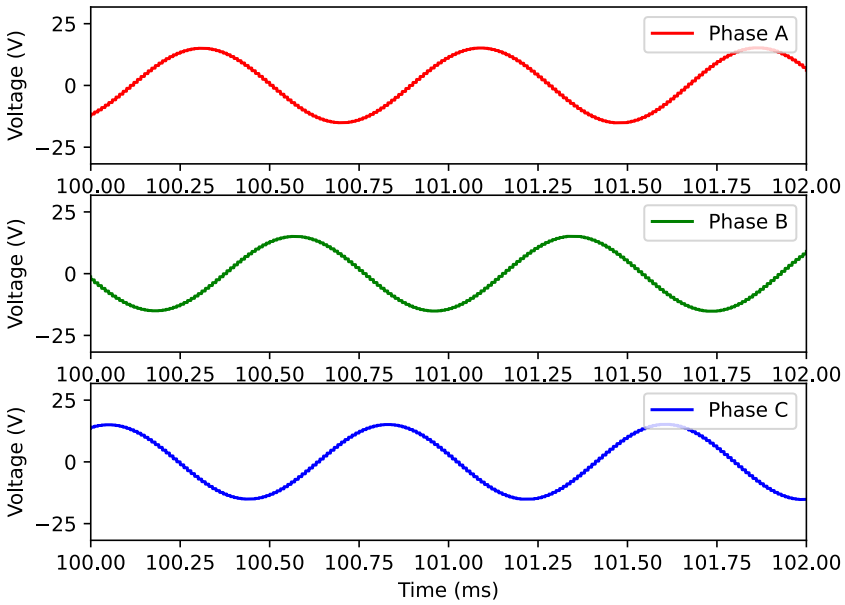

Figure I.45: Zoomed in phase voltages

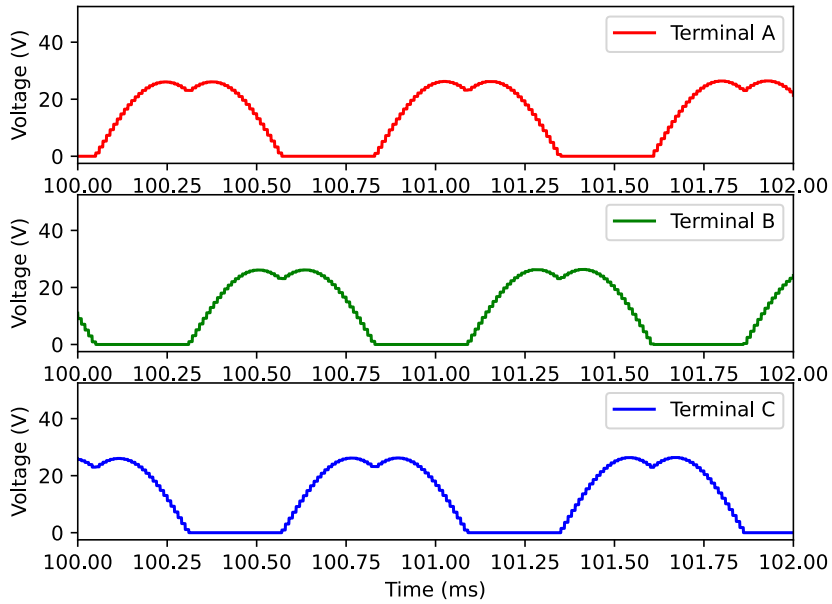

Figure I.46: Zoomed in terminal voltages

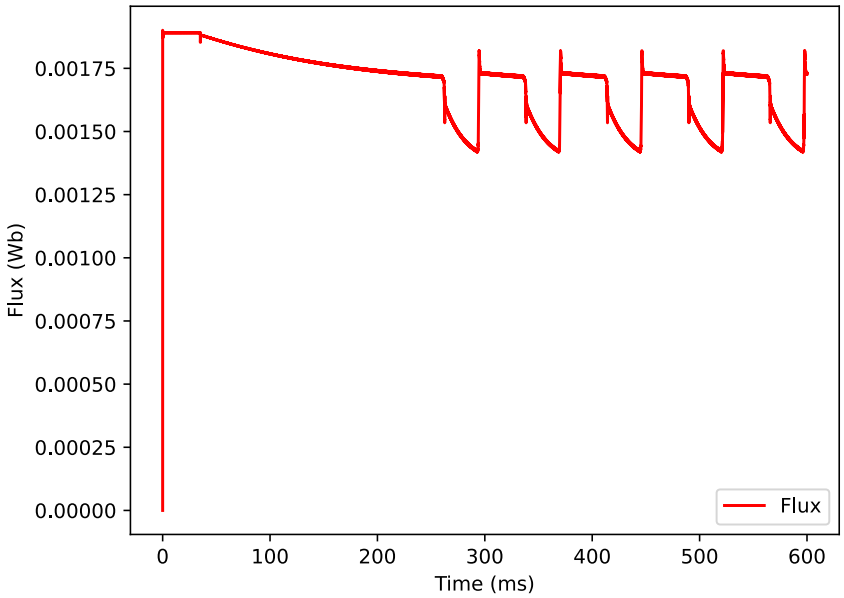

Figure I.47: Flux generated by the motor, for more information see code

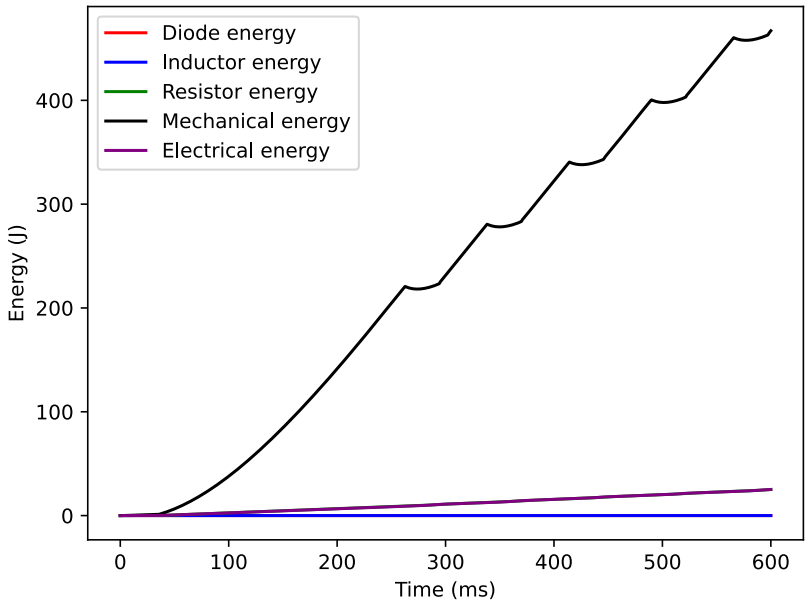

Figure I.48: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.949

Various starting velocities

Trapezoidal control

0 rad/s

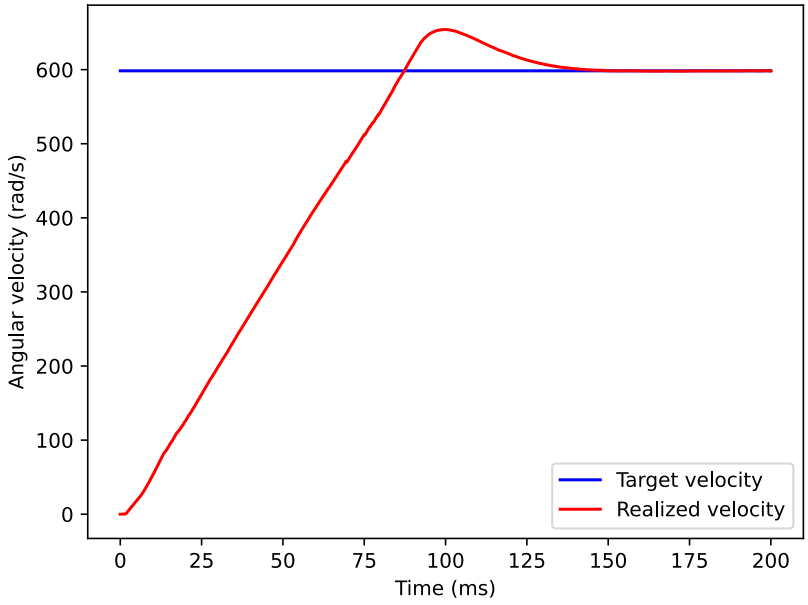

Figure I.49: The desired and realized velocity

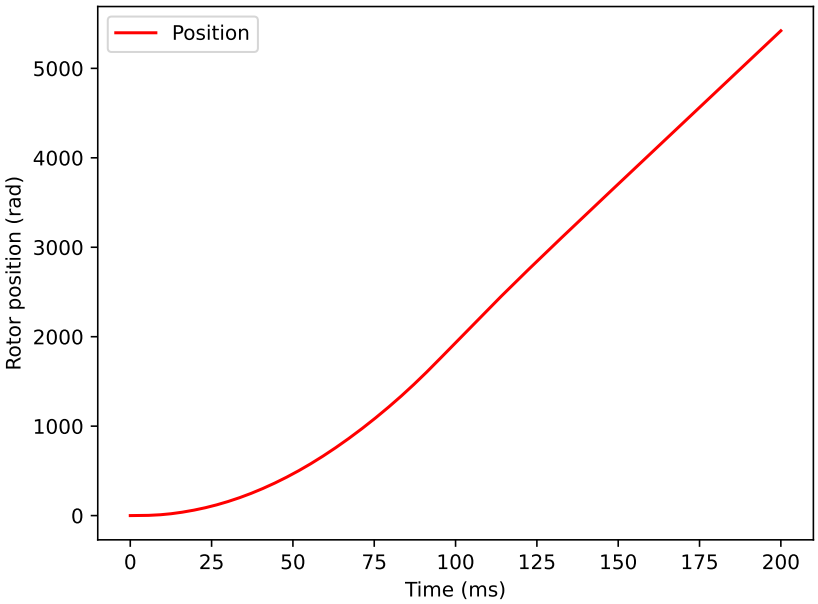

Figure I.50: The position of the rotor

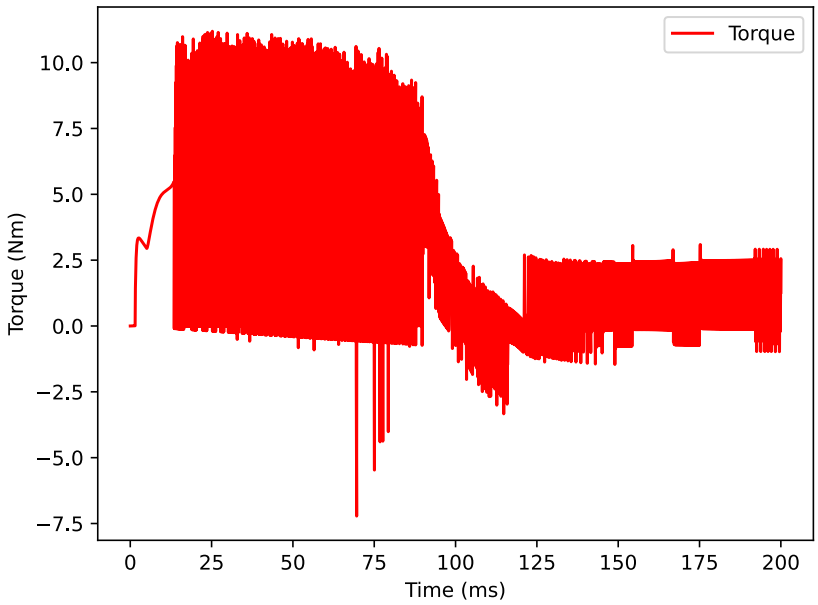

Figure I.51: The torque delivered by the motor

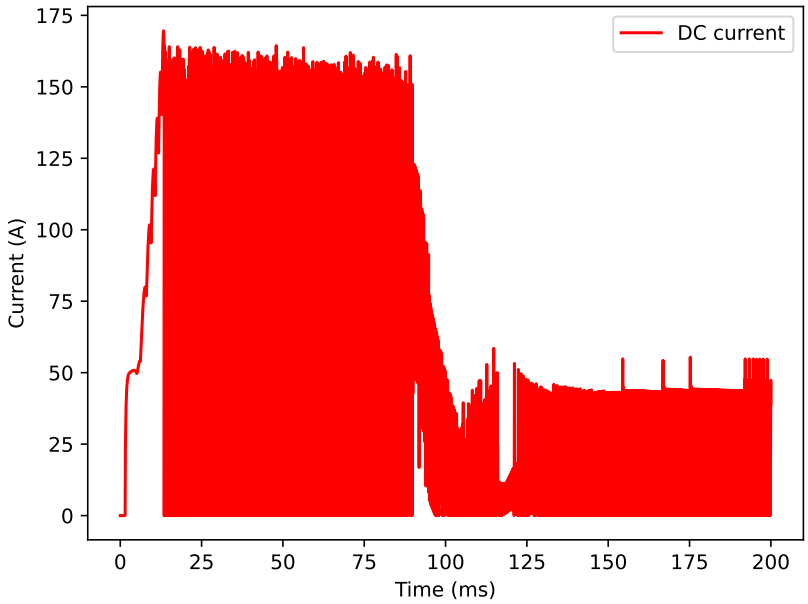

Figure I.52: The DC current through the motor

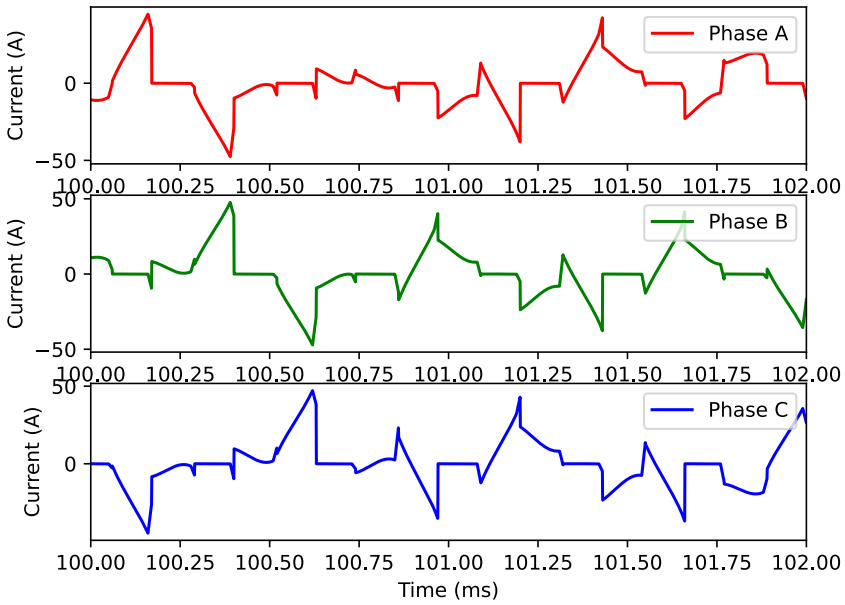

Figure I.53: Zoomed in phase currents through the motor

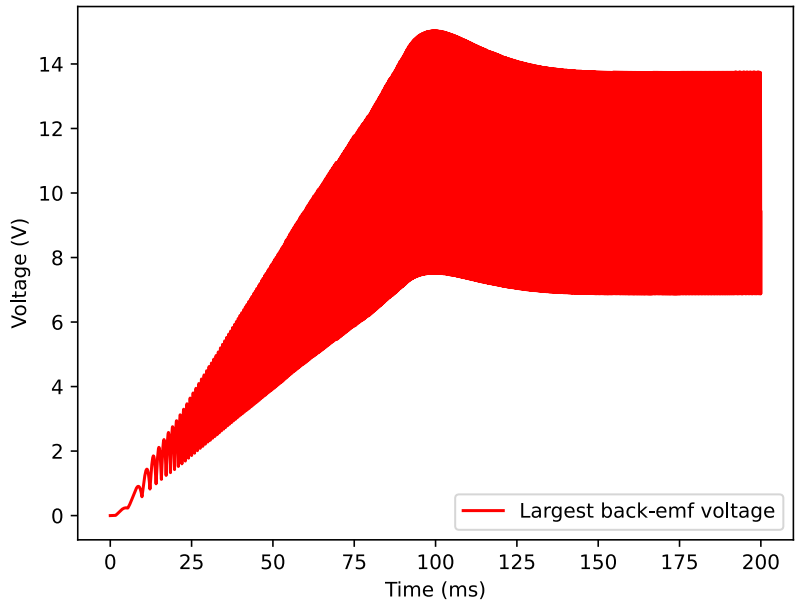

Figure I.54: The largest value for back-emf generated by a motor phase

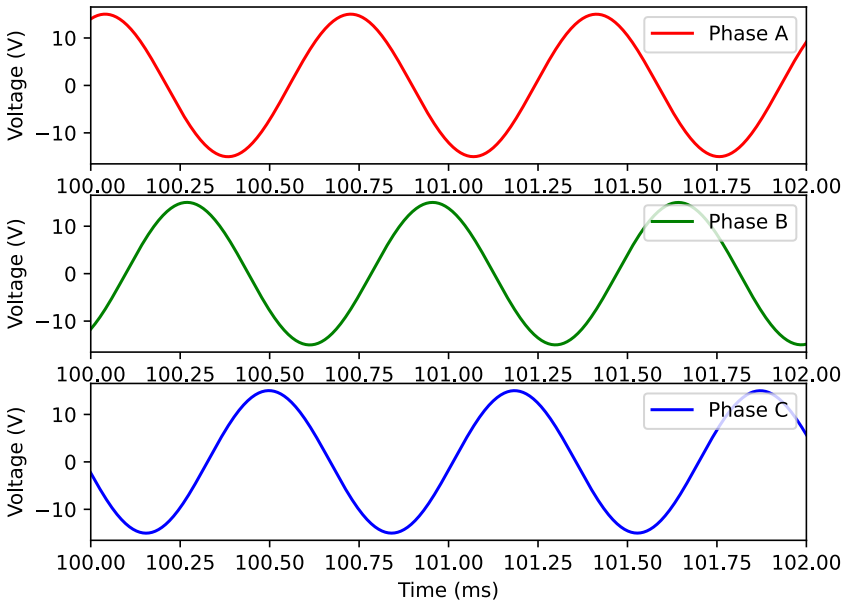

Figure I.55: Zoomed in back-emf generated by each phase

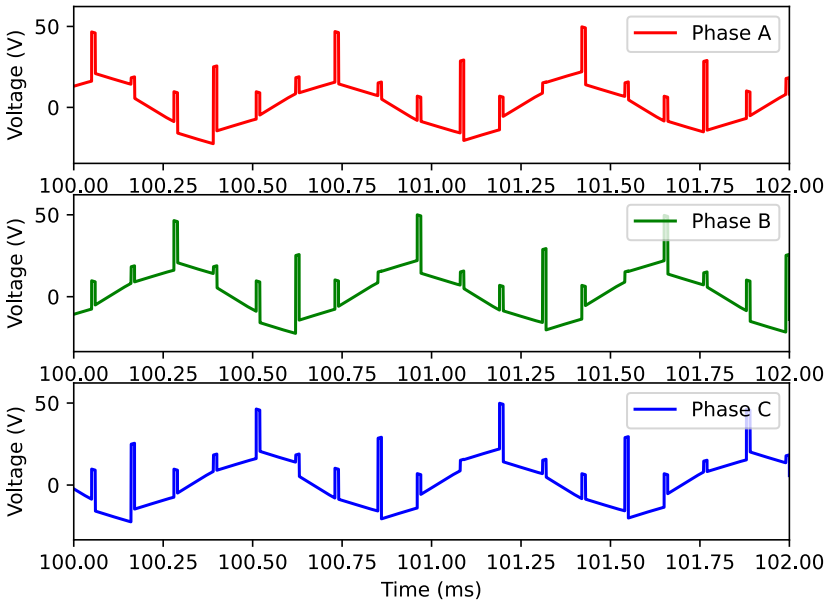

Figure I.56: Zoomed in phase voltages

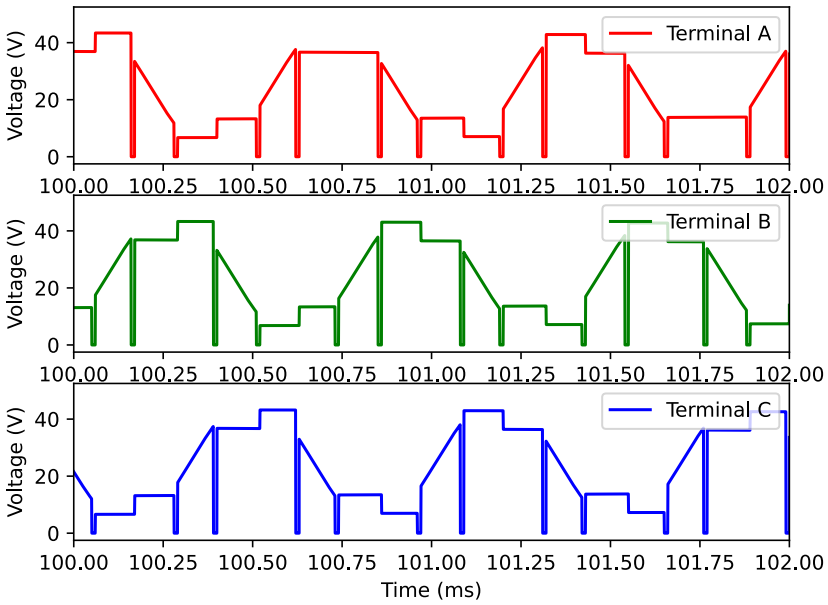

Figure I.57: Zoomed in terminal voltages

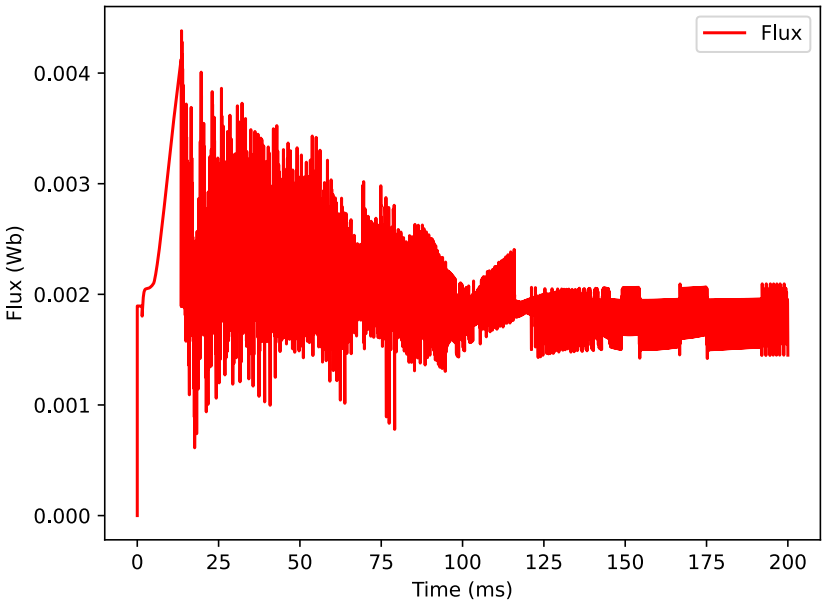

Figure I.58: Flux generated by the motor, for more information see code

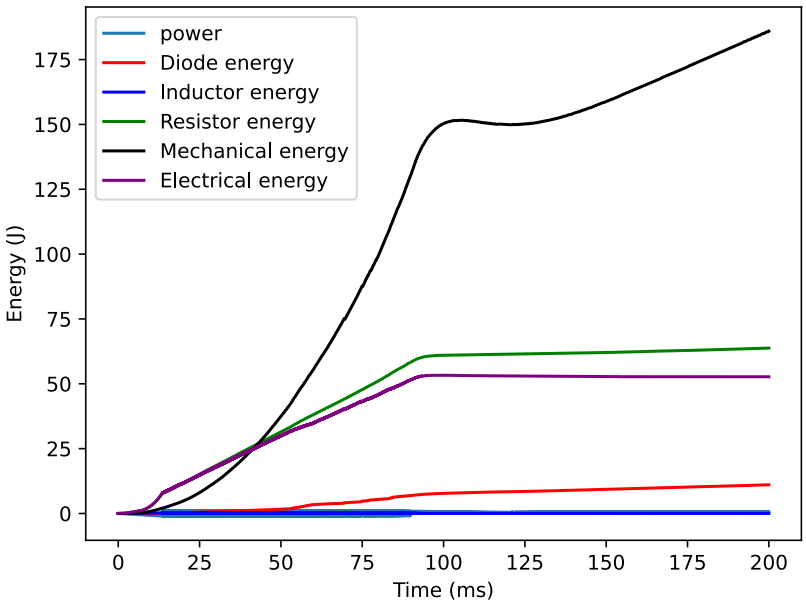

Figure I.59: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.779

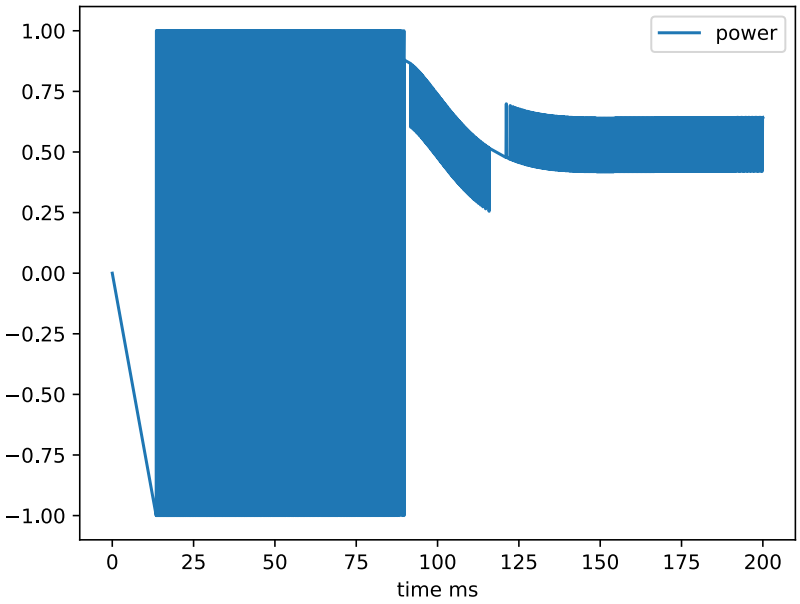

Figure I.60: The amplitude of the normalized output voltage

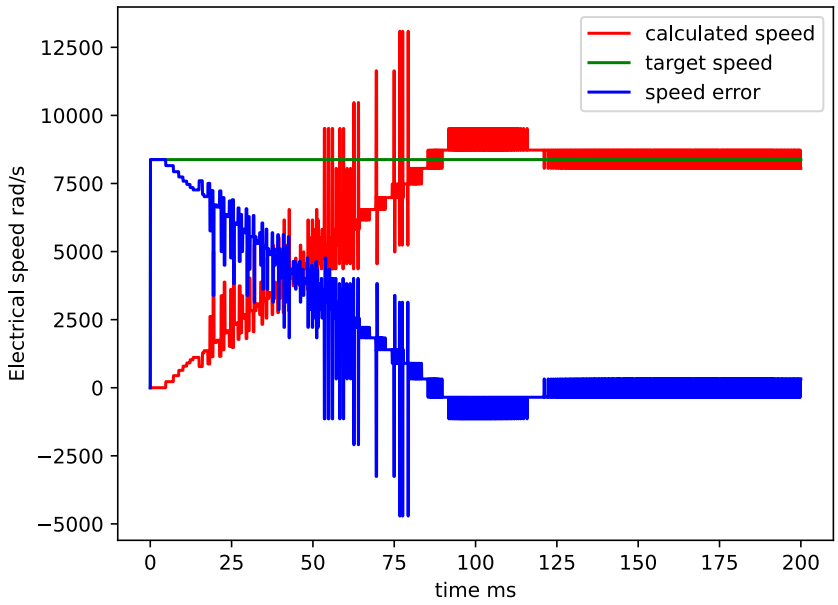

Figure I.61: The calculated, target, and error in the electrical speed of the trapezoidal controller

300 rad/s

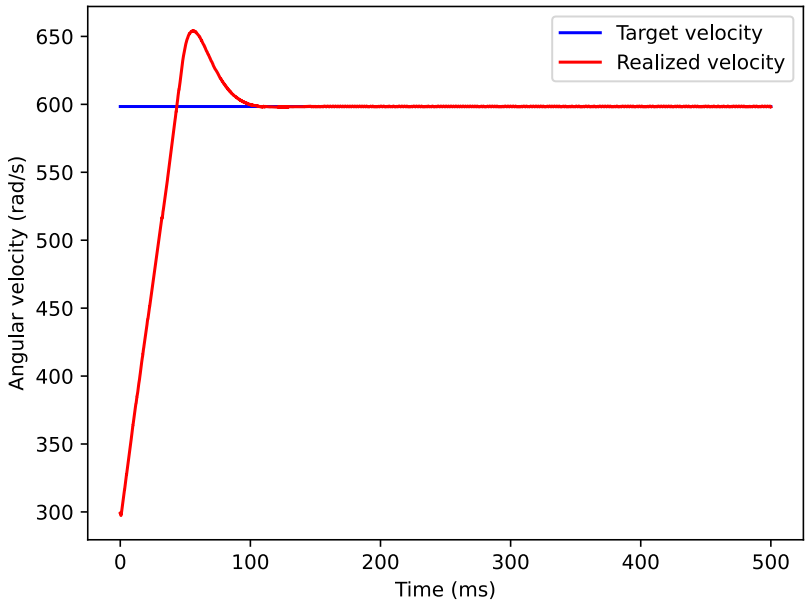

Figure I.62: The desired and realized velocity

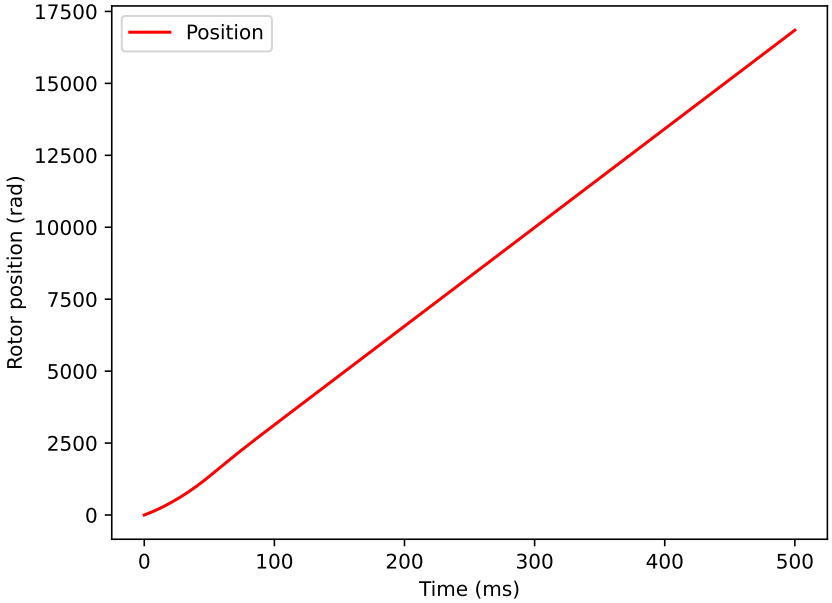

Figure I.63: The position of the rotor

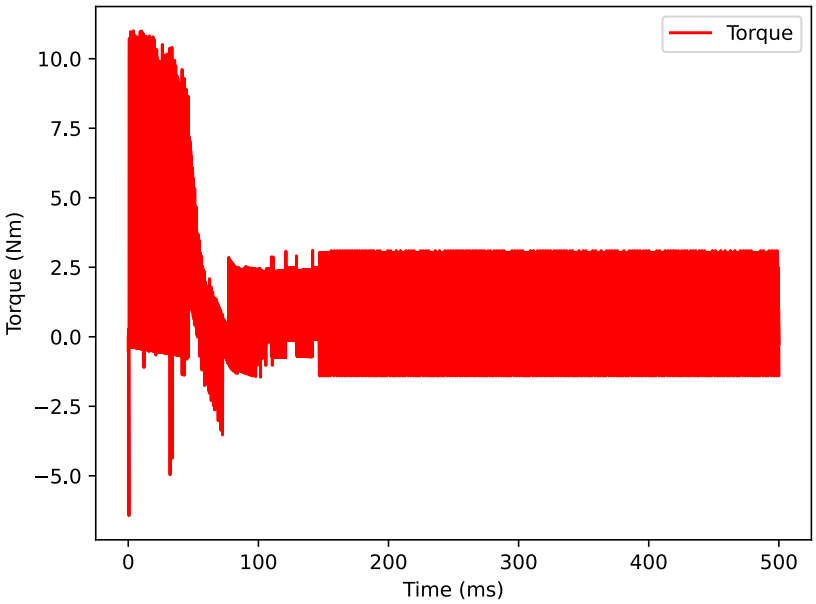

Figure I.64: The torque delivered by the motor

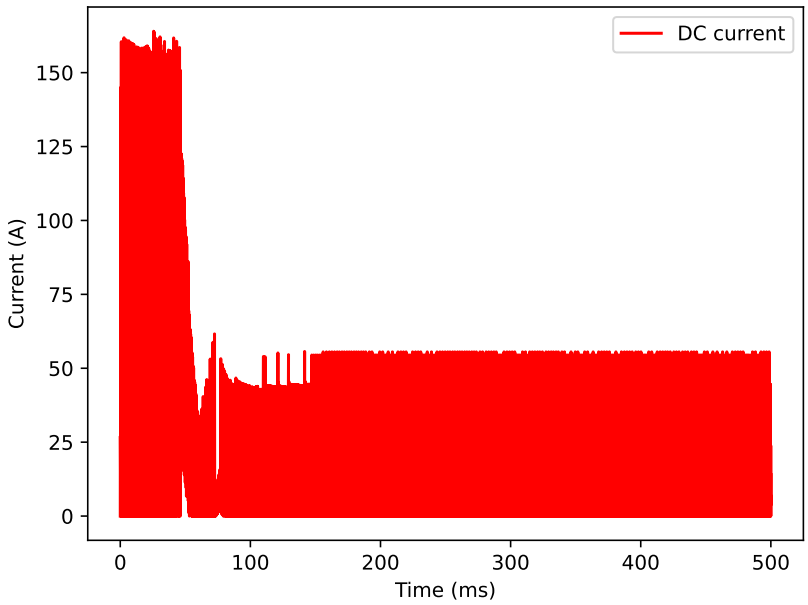

Figure I.65: The DC current through the motor

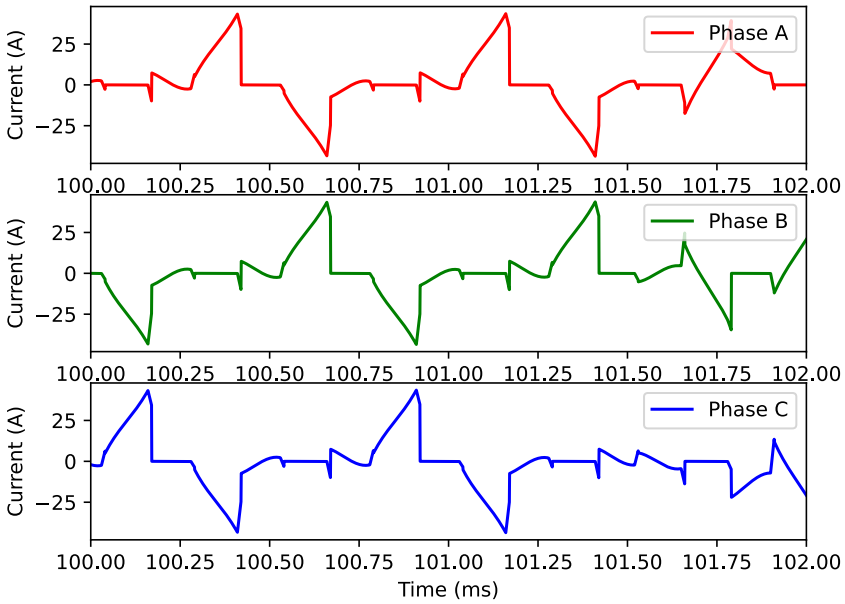

Figure I.66: Zoomed in phase currents through the motor

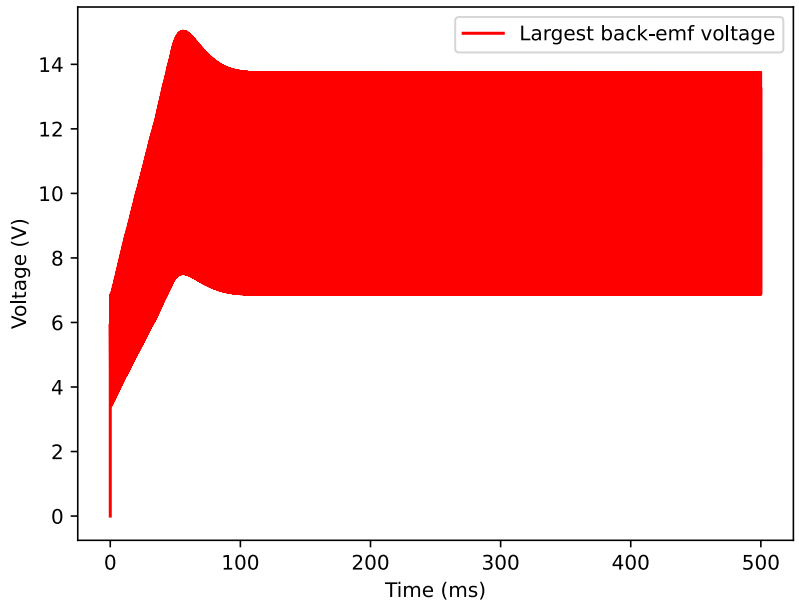

Figure I.67: The largest value for back-emf generated by a motor phase

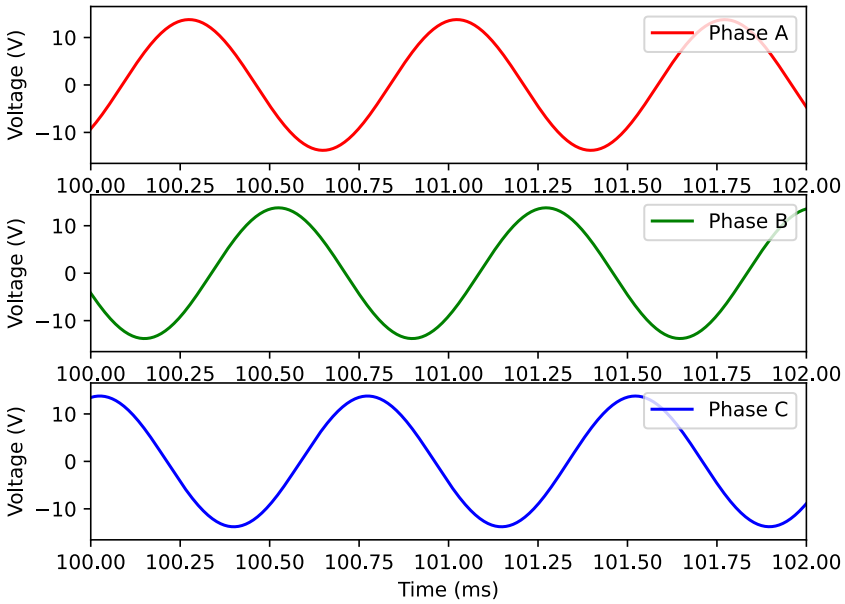

Figure I.68: Zoomed in back-emf generated by each phase

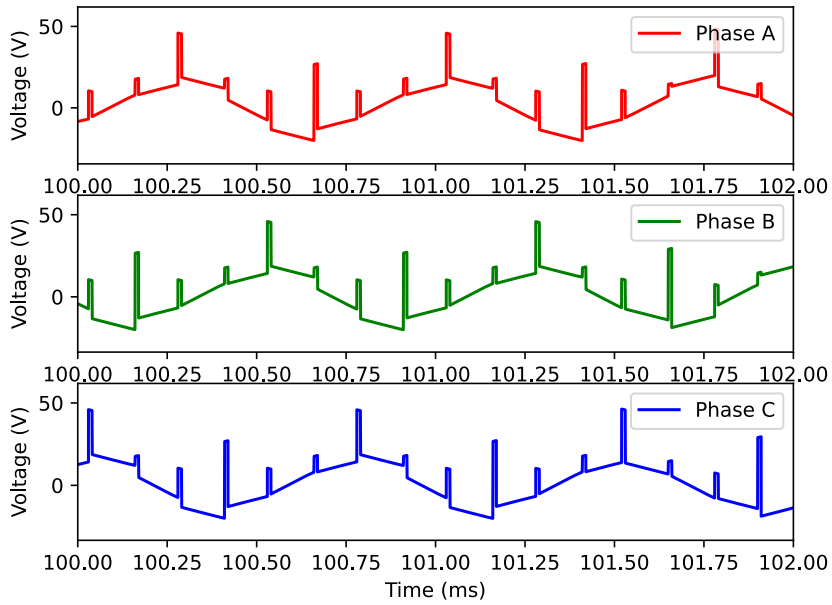

Figure I.69: Zoomed in phase voltages

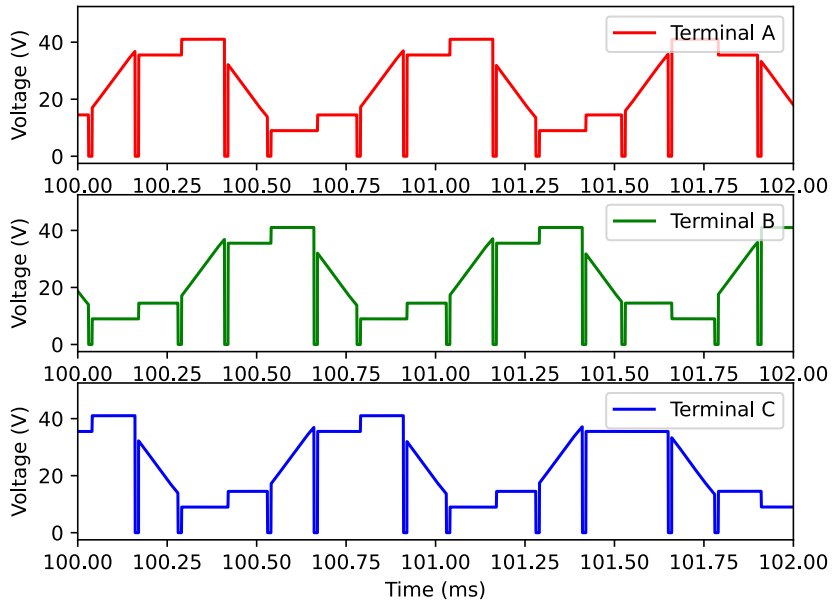

Figure I.70: Zoomed in terminal voltages

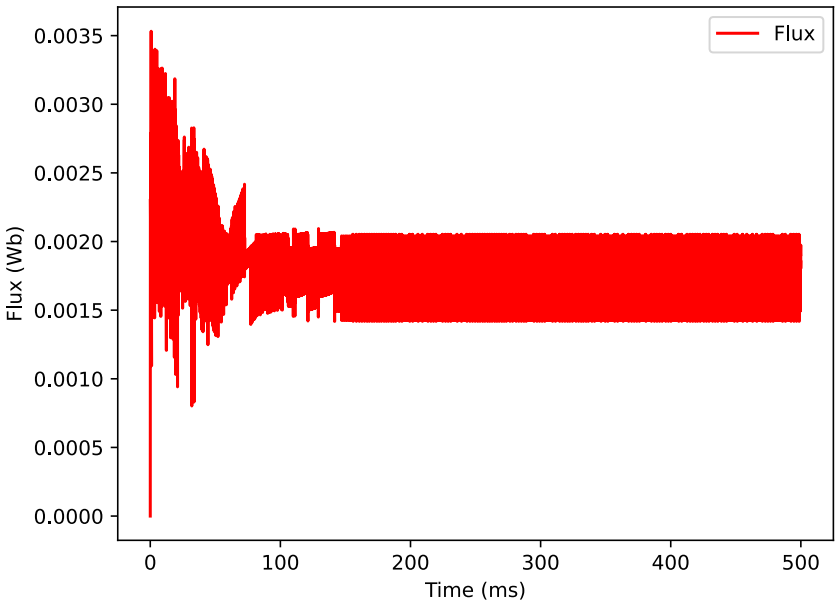

Figure I.71: Flux generated by the motor, for more information see code

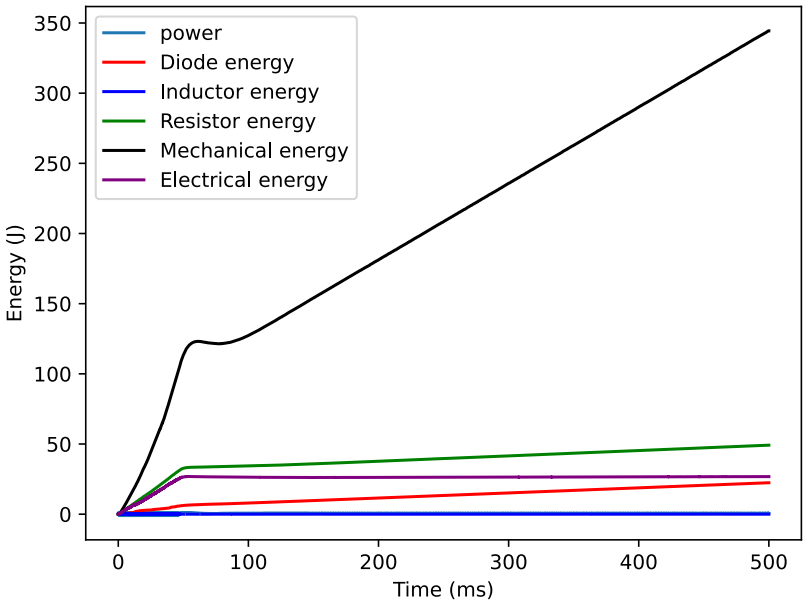

Figure I.72: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.928

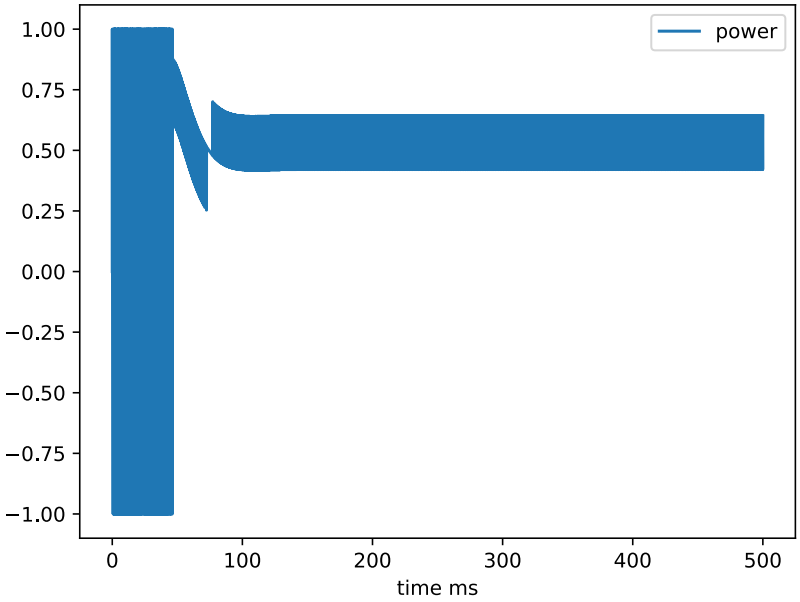

Figure I.73: The amplitude of the normalized output voltage

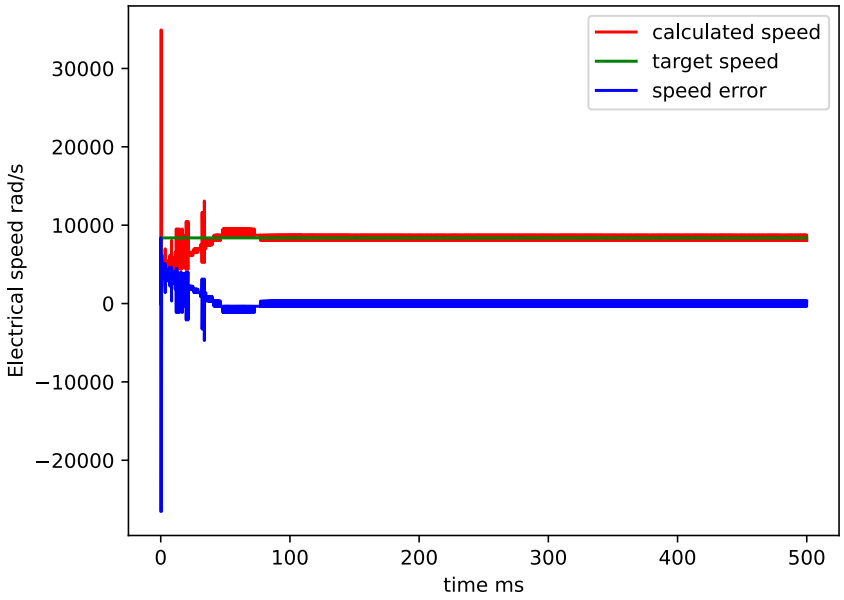

Figure I.74: The calculated, target, and error in the electrical speed of the trapezoidal controller

900 rad/s

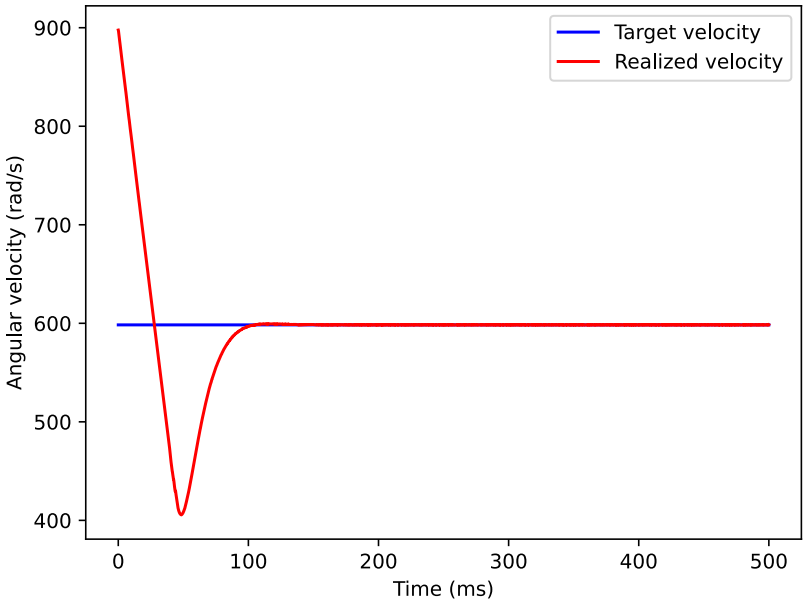

Figure I.75: The desired and realized velocity

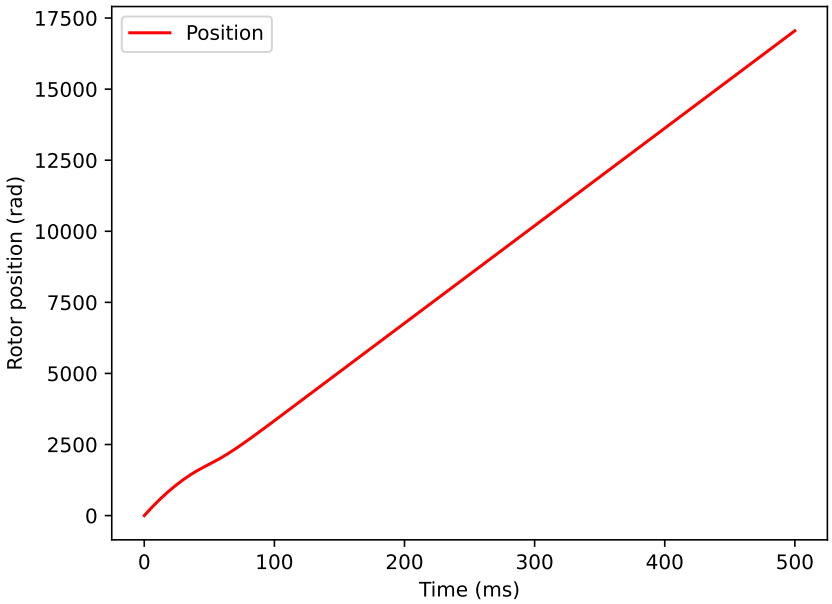

Figure I.76: The position of the rotor

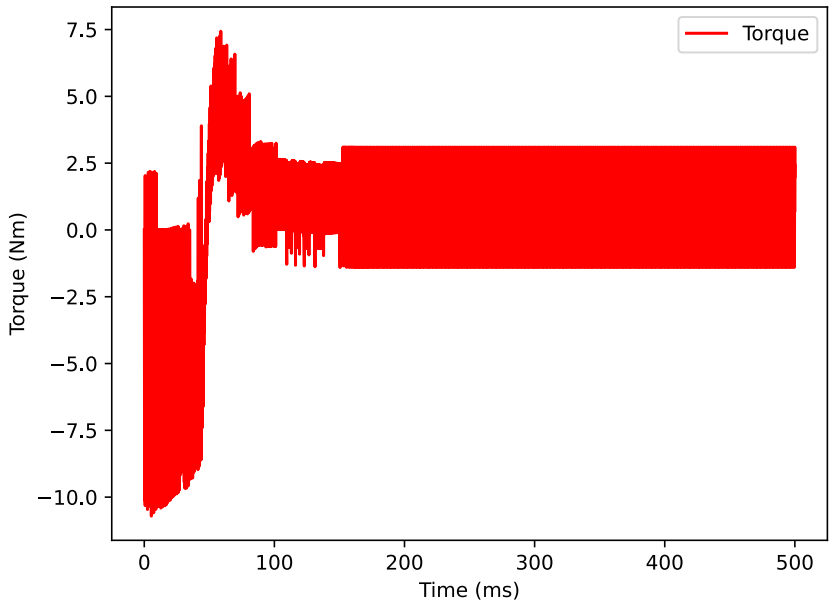

Figure I.77: The torque delivered by the motor

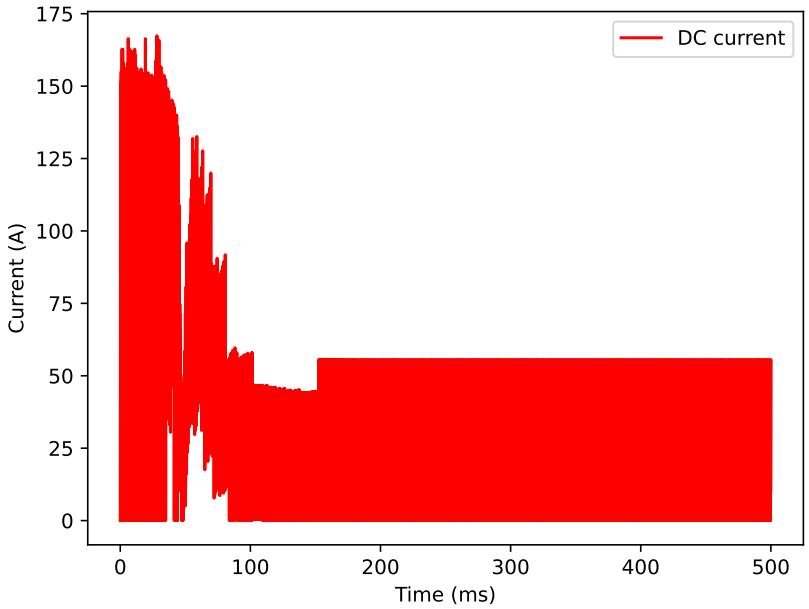

Figure I.78: The DC current through the motor

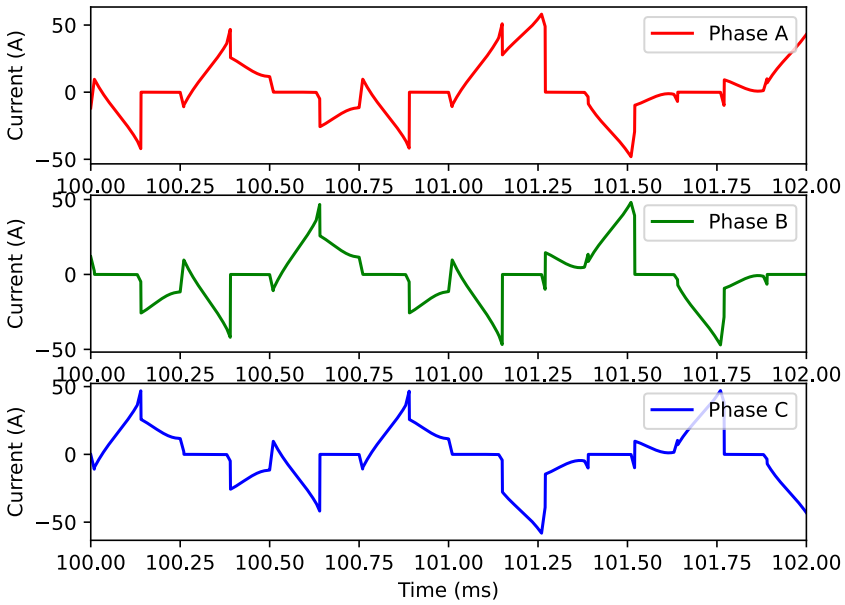

Figure I.79: Zoomed in phase currents through the motor

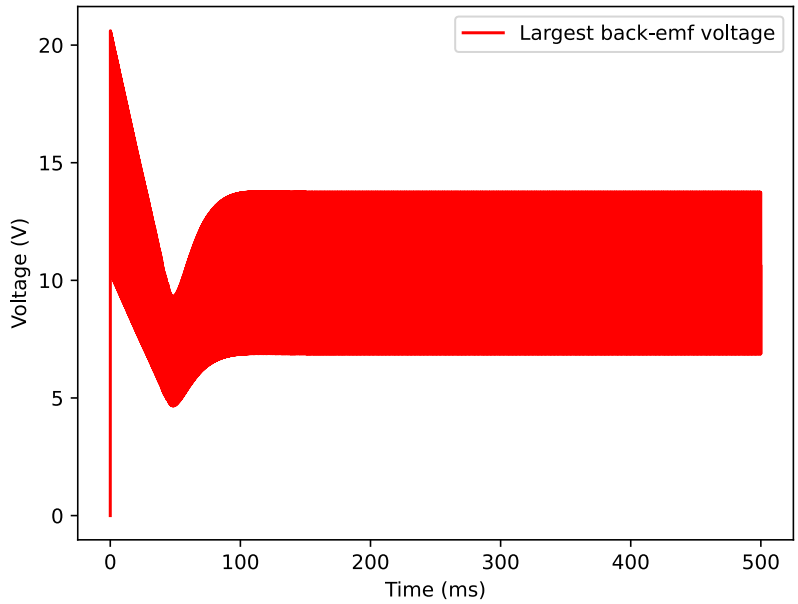

Figure I.80: The largest value for back-emf generated by a motor phase

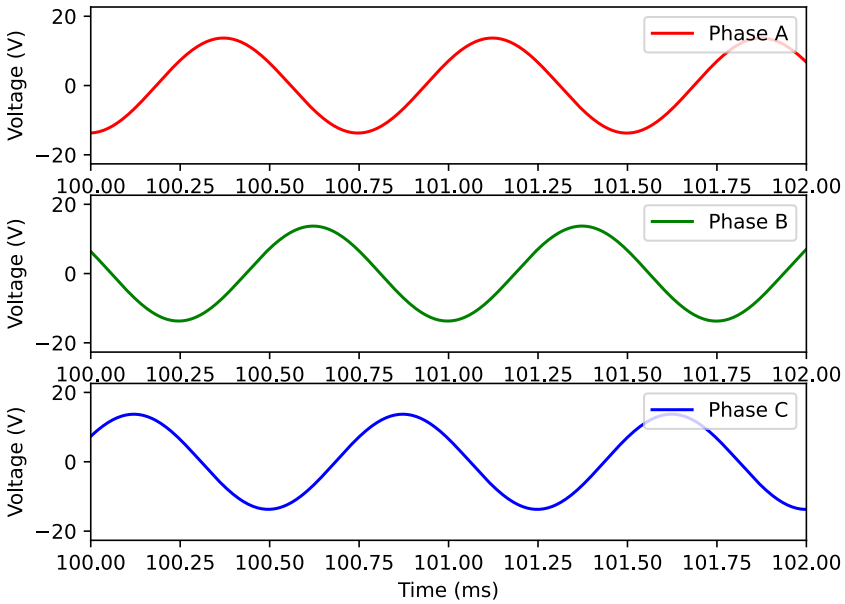

Figure I.81: Zoomed in back-emf generated by each phase

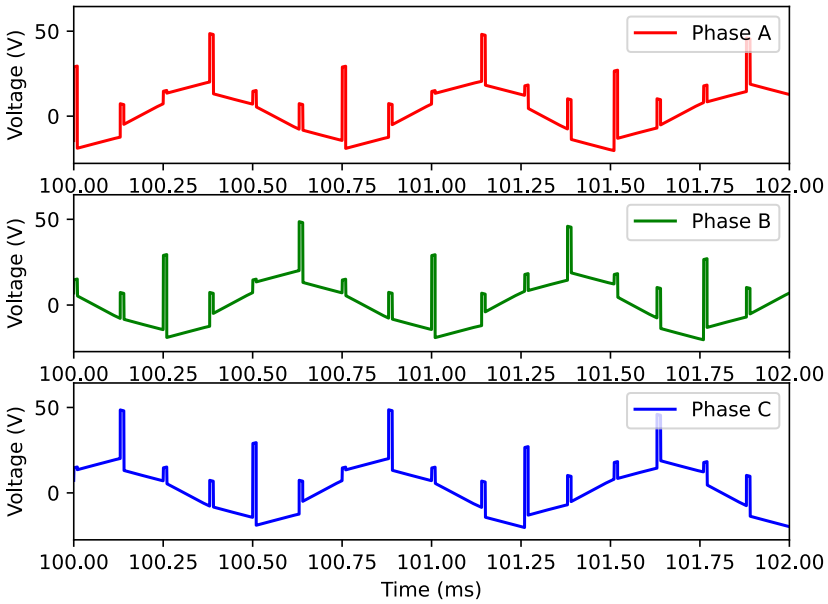

Figure I.82: Zoomed in phase voltages

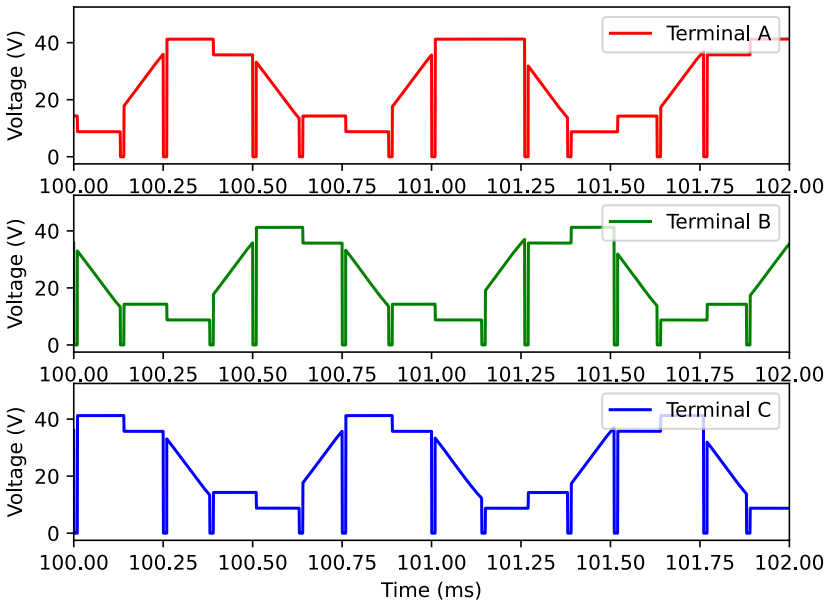

Figure I.83: Zoomed in terminal voltages

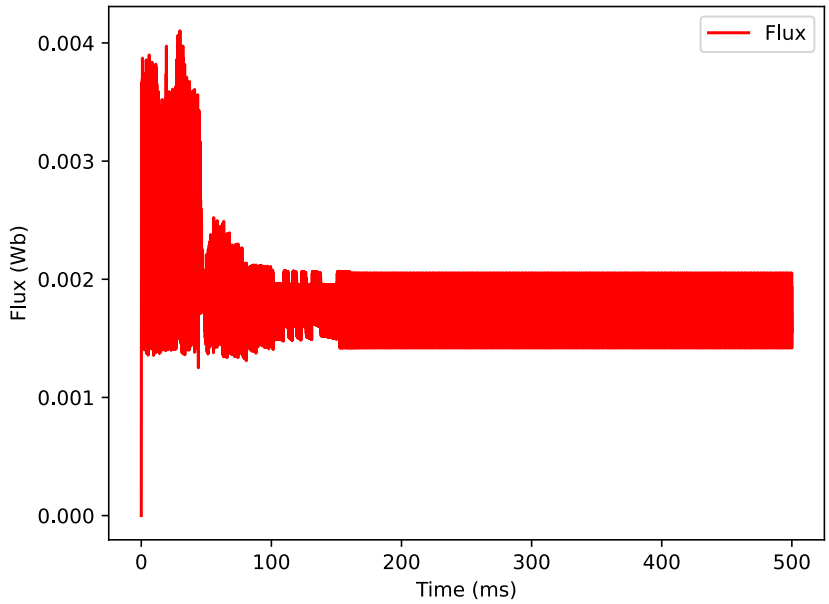

Figure I.84: Flux generated by the motor, for more information see code

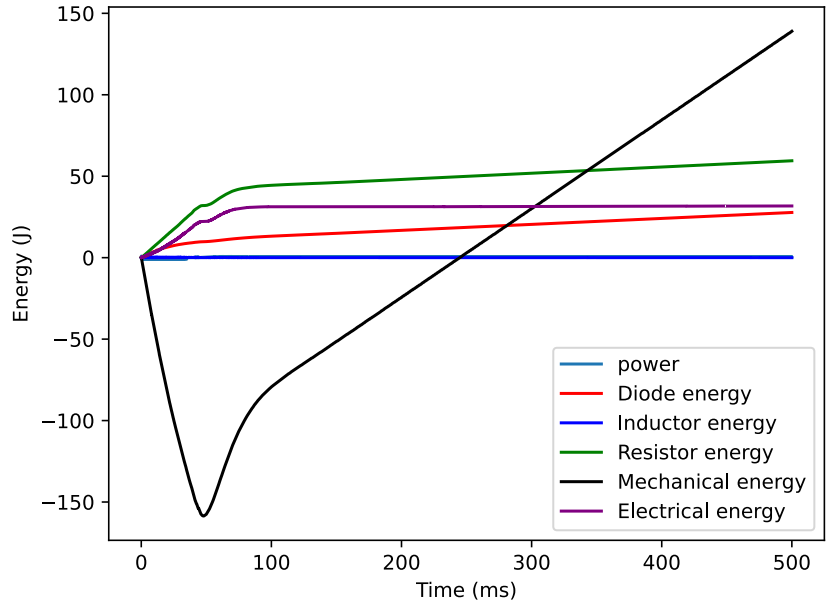

Figure I.85: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.814

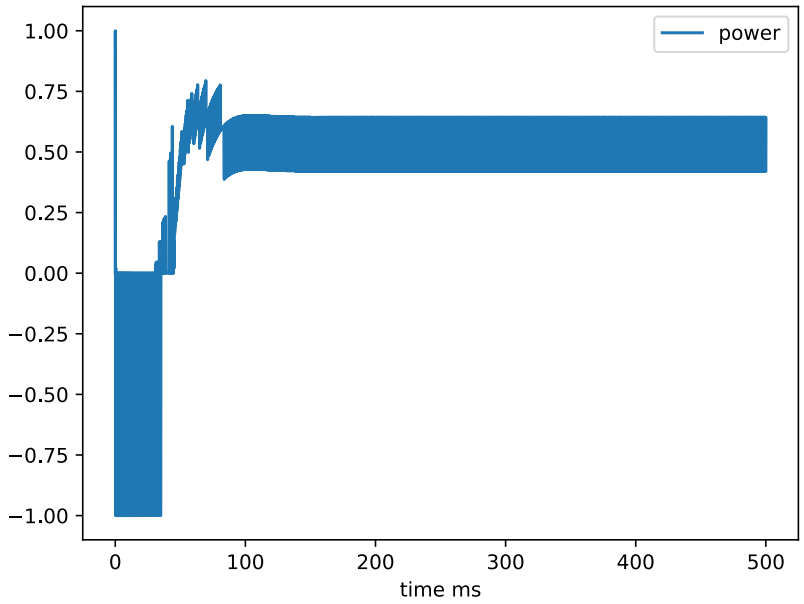

Figure I.86: The amplitude of the normalized output voltage

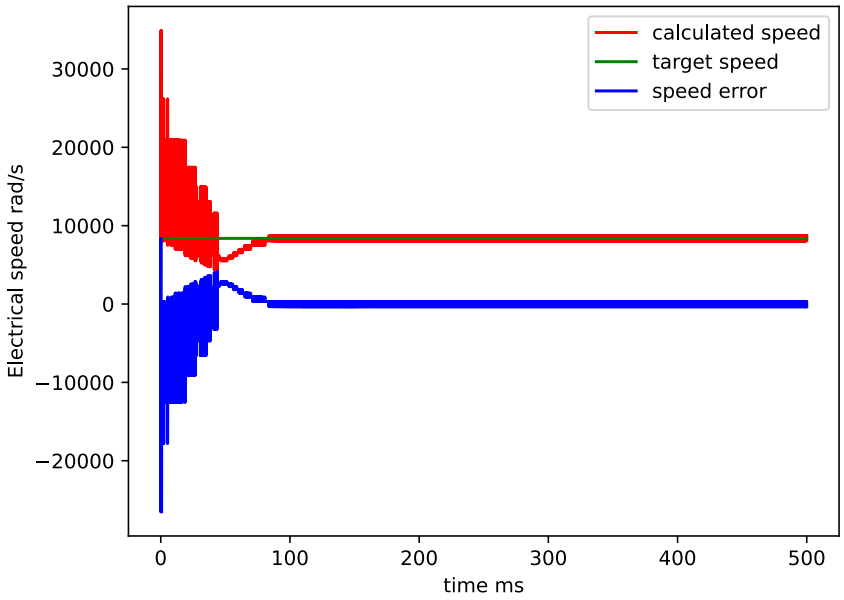

Figure I.87: The calculated, target, and error in the electrical speed of the trapezoidal controller

FOC

0 rad/s

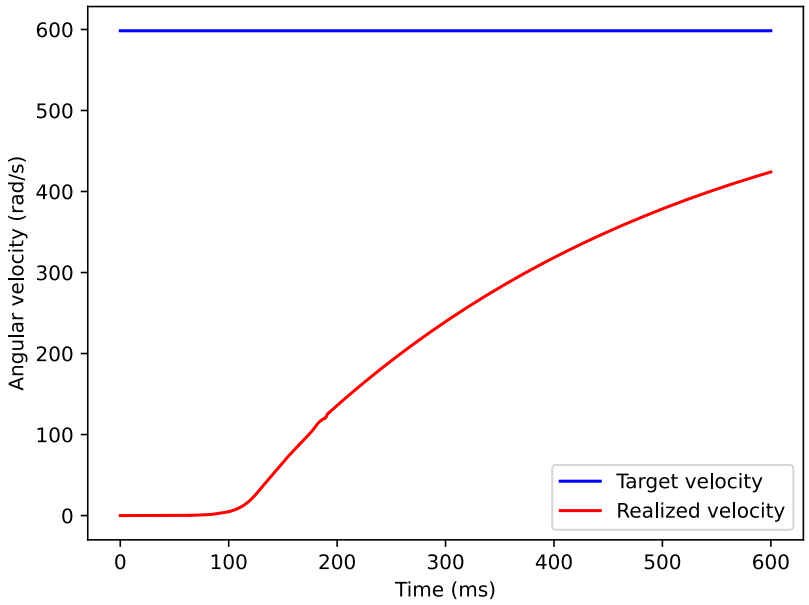

Figure I.88: The desired and realized velocity

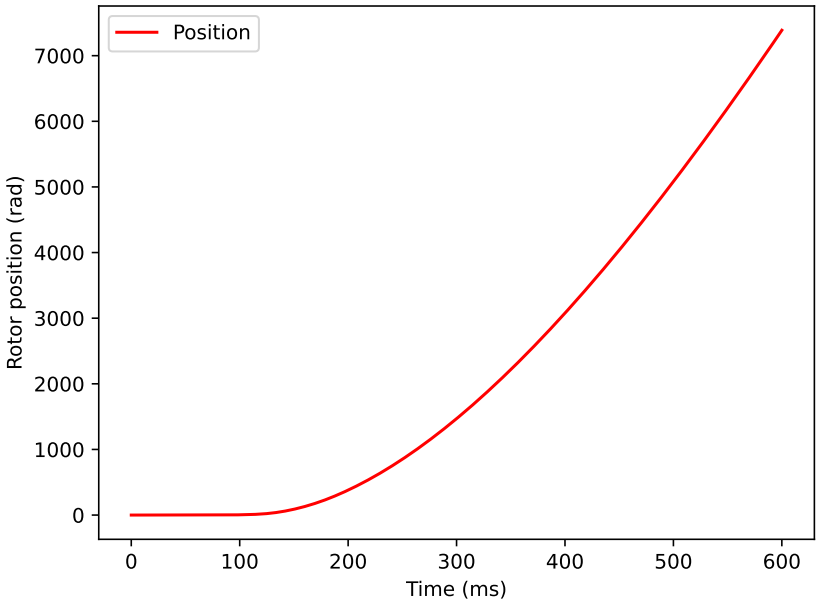

Figure I.89: The position of the rotor

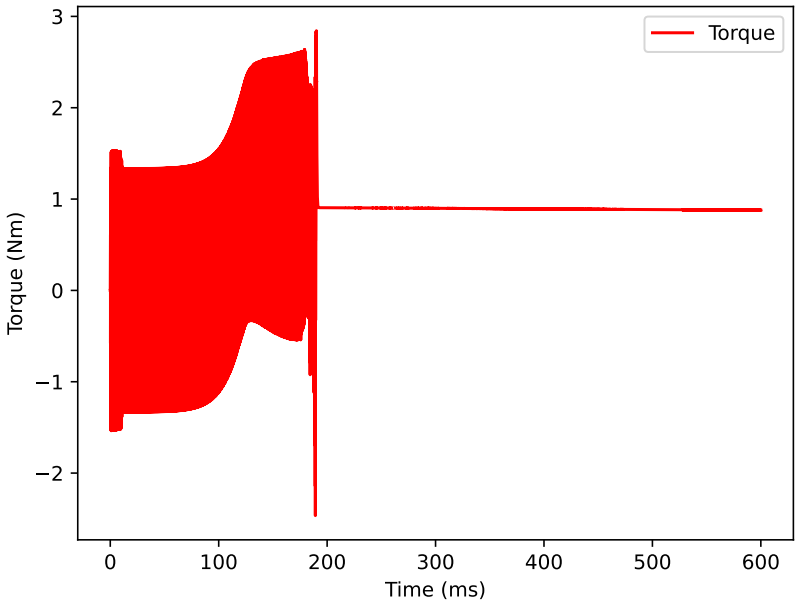

Figure I.90: The torque delivered by the motor

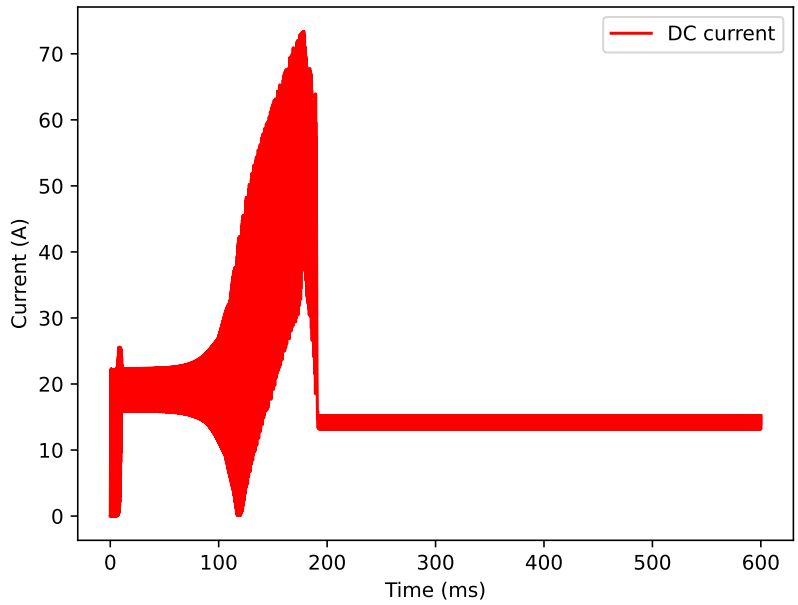

Figure I.91: The DC current through the motor

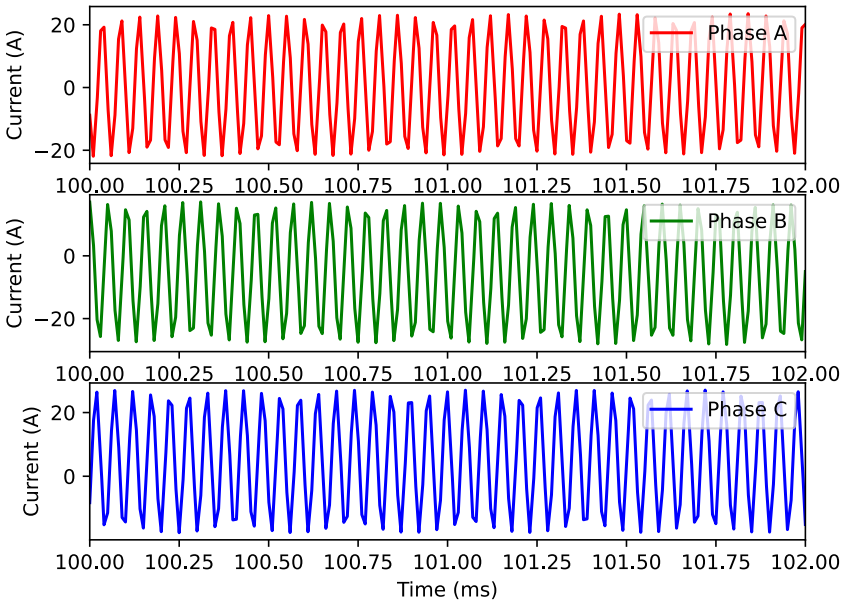

Figure I.92: Zoomed in phase currents through the motor

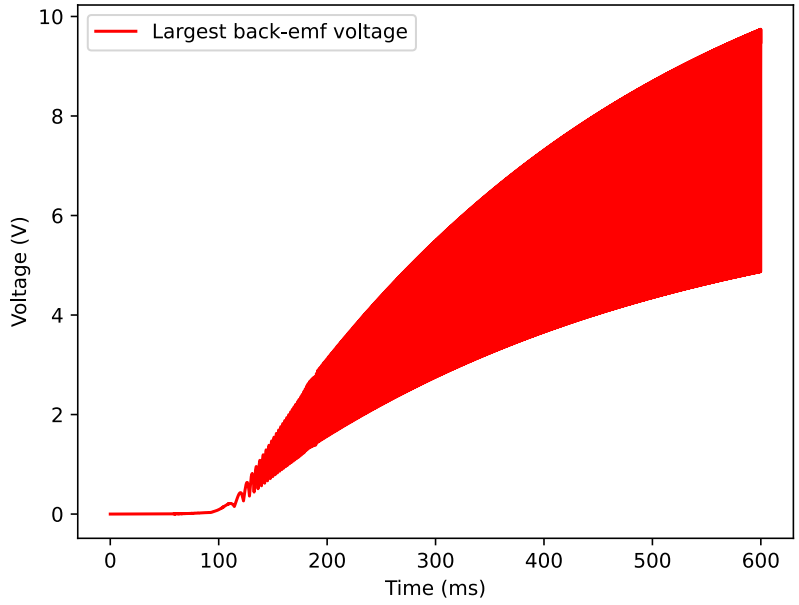

Figure I.93: The largest value for back-emf generated by a motor phase

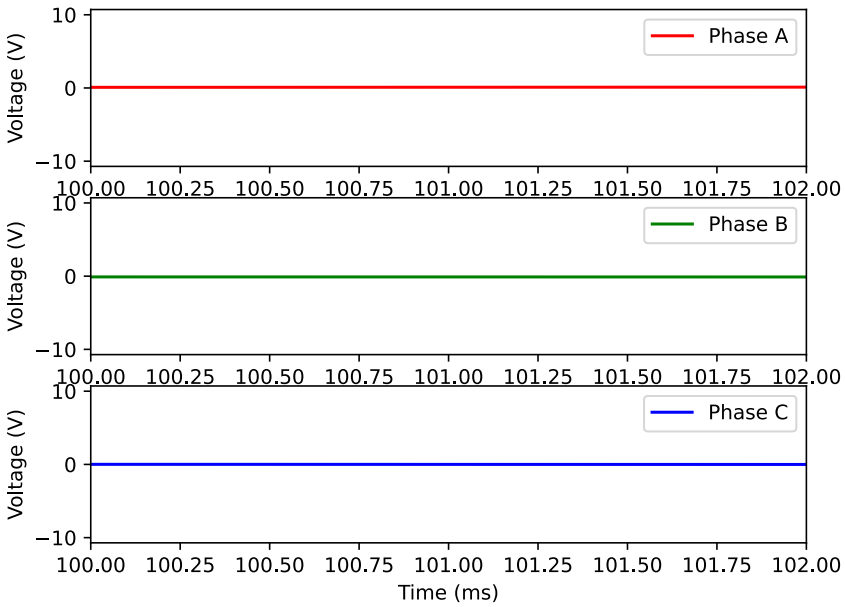

Figure I.94: Zoomed in back-emf generated by each phase

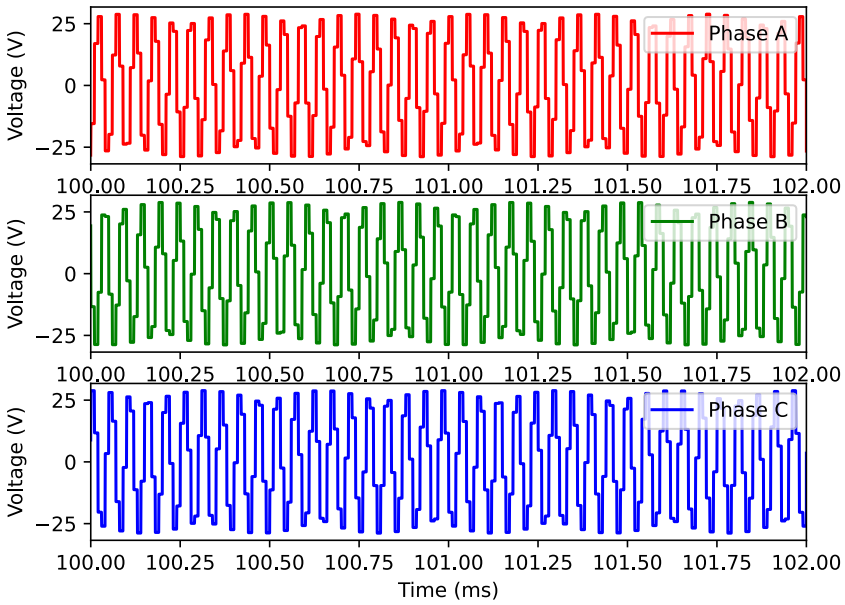

Figure I.95: Zoomed in phase voltages

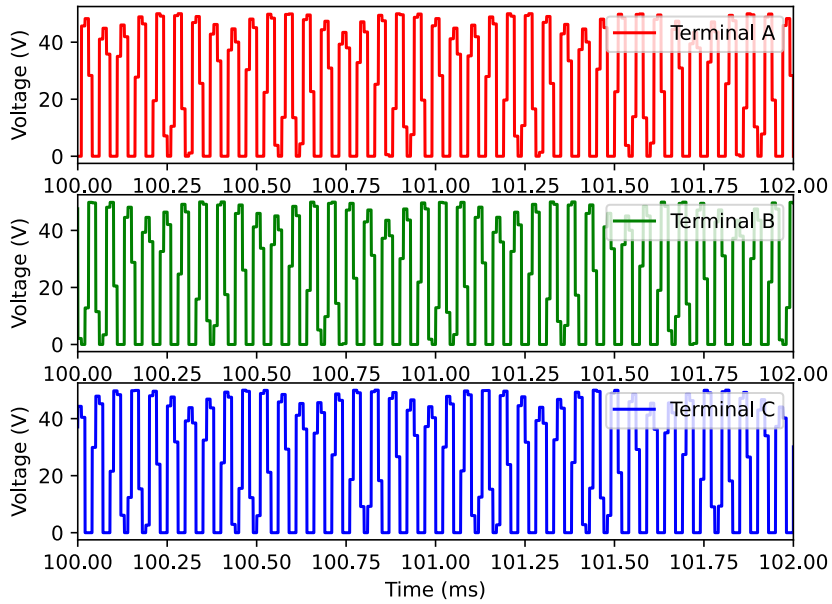

Figure I.96: Zoomed in terminal voltages

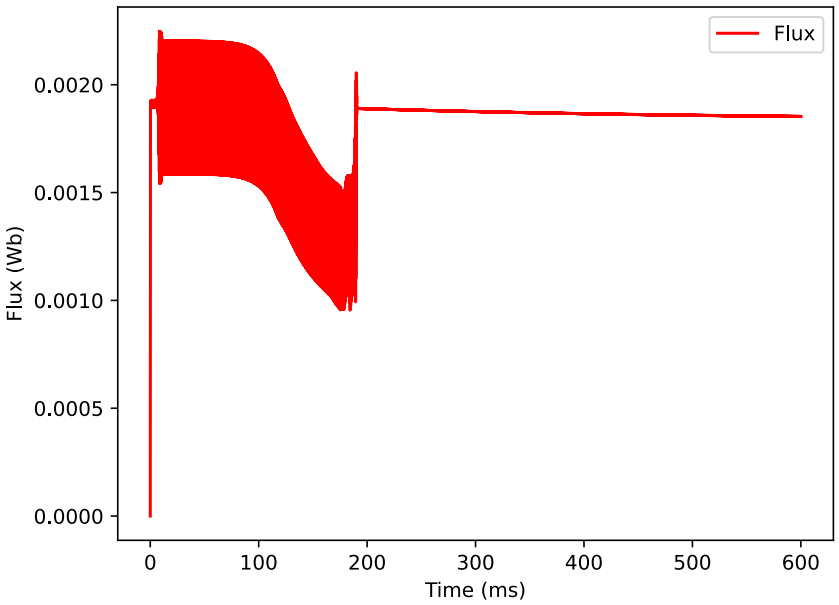

Figure I.97: Flux generated by the motor, for more information see code

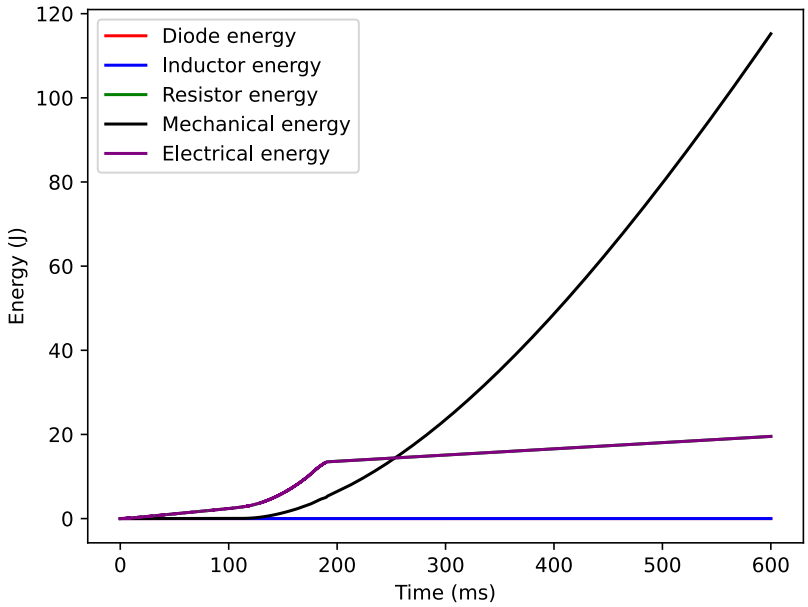

Figure I.98: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.855

300 rad/s

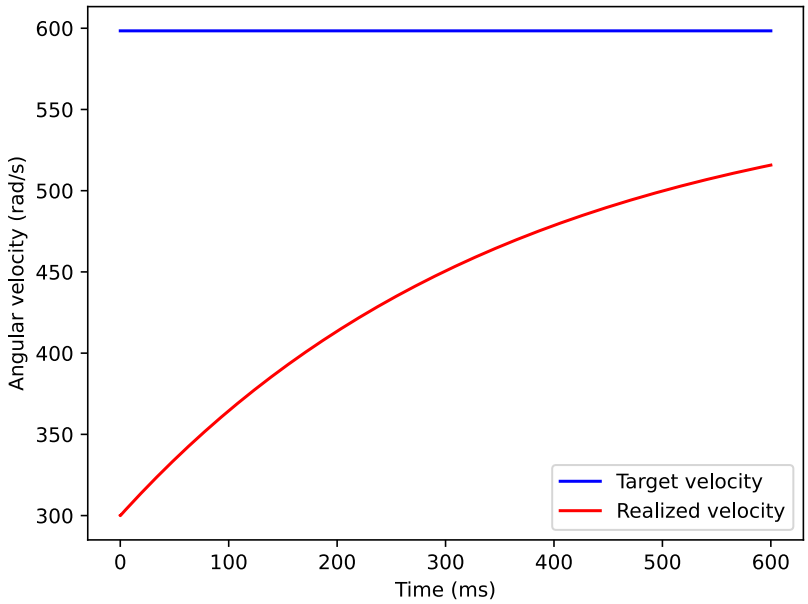

Figure I.99: The desired and realized velocity

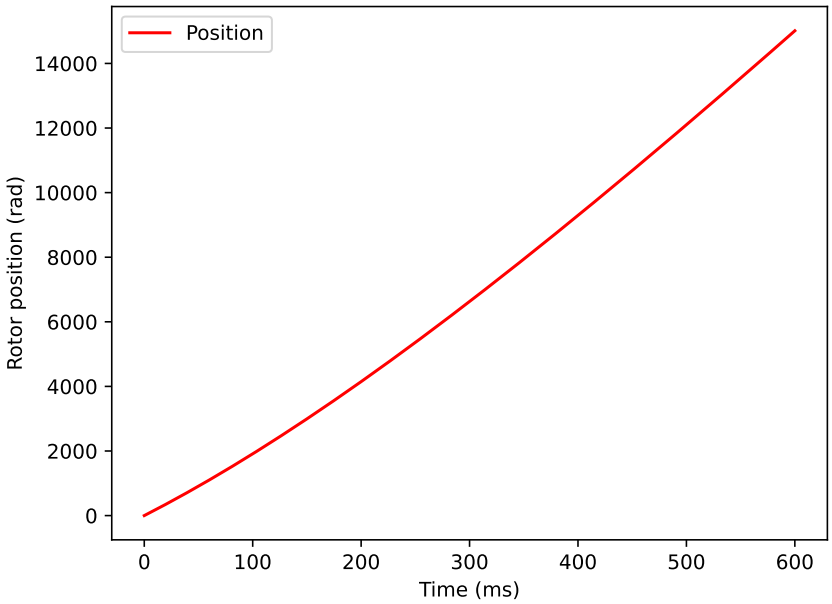

Figure I.100: The position of the rotor

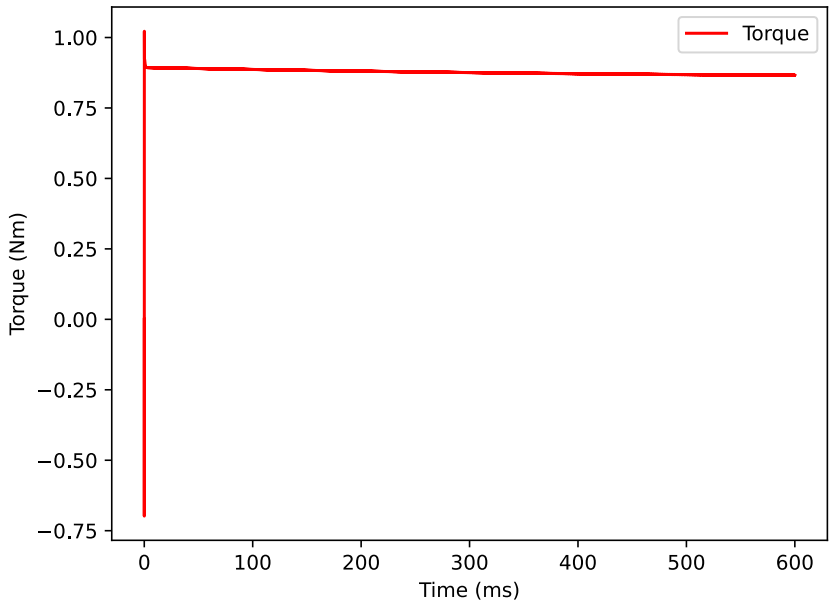

Figure I.101: The torque delivered by the motor

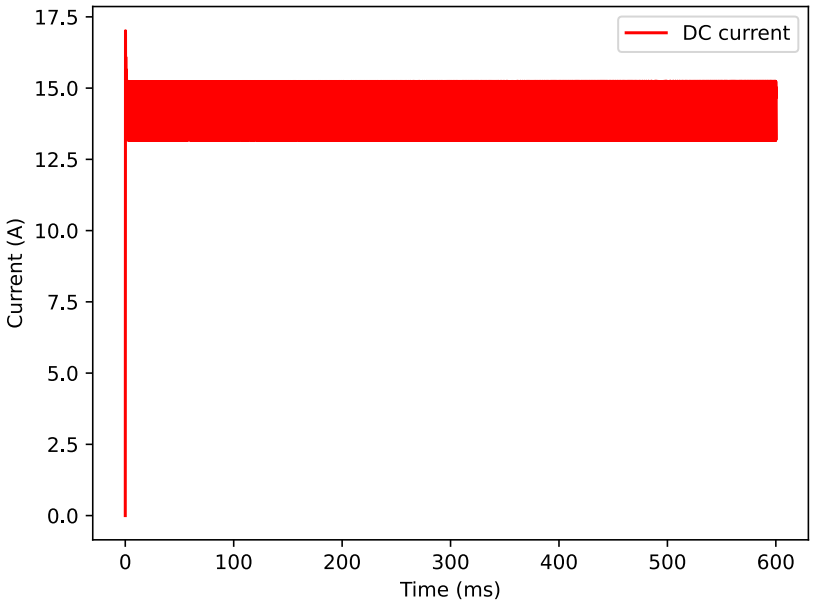

Figure I.102: The DC current through the motor

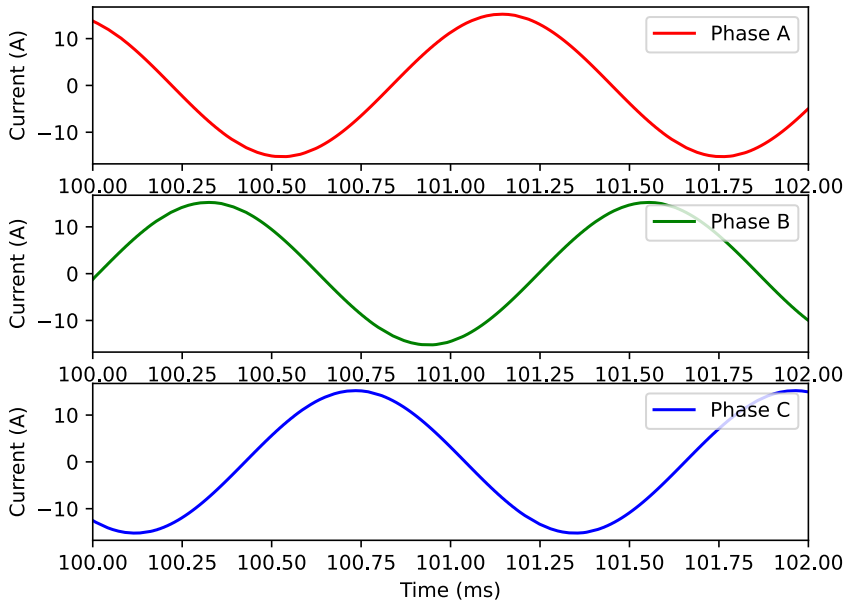

Figure I.103: Zoomed in phase currents through the motor

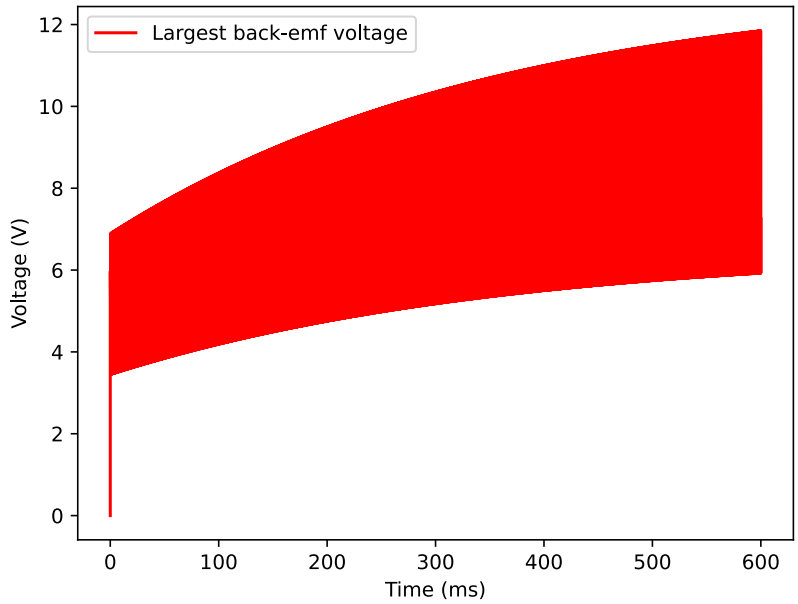

Figure I.104: The largest value for back-emf generated by a motor phase

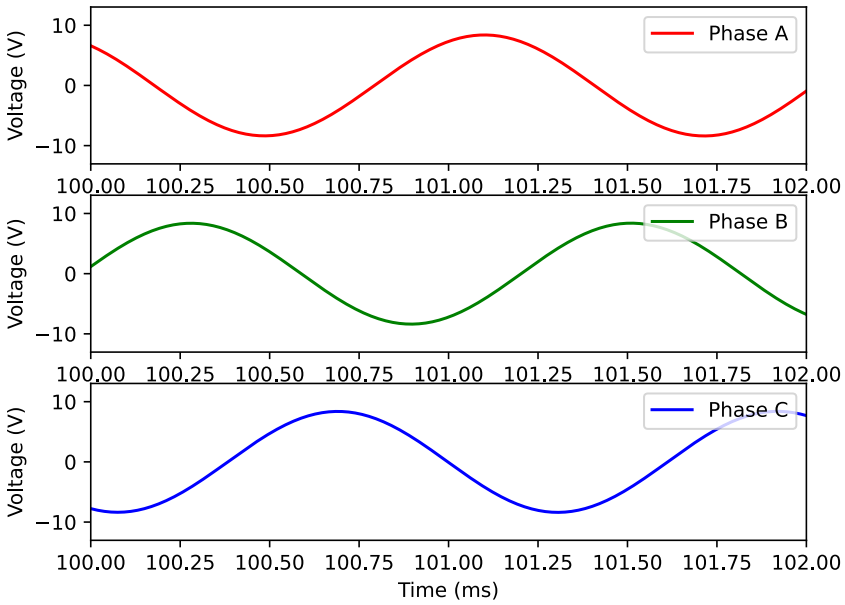

Figure I.105: Zoomed in back-emf generated by each phase

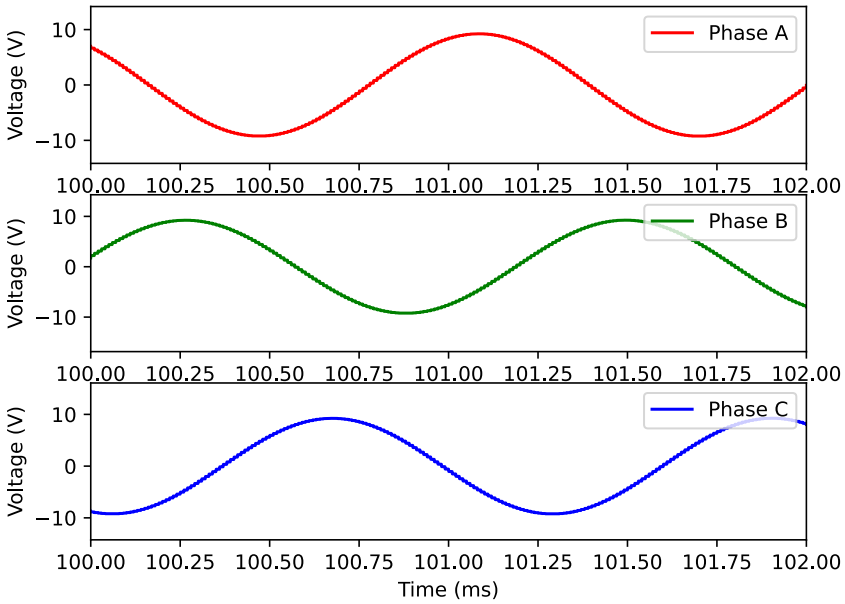

Figure I.106: Zoomed in phase voltages

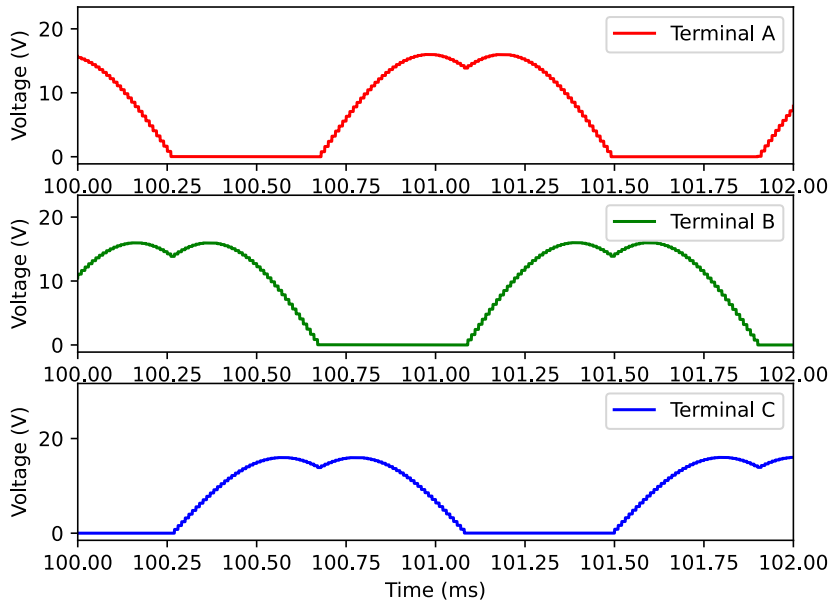

Figure I.107: Zoomed in terminal voltages

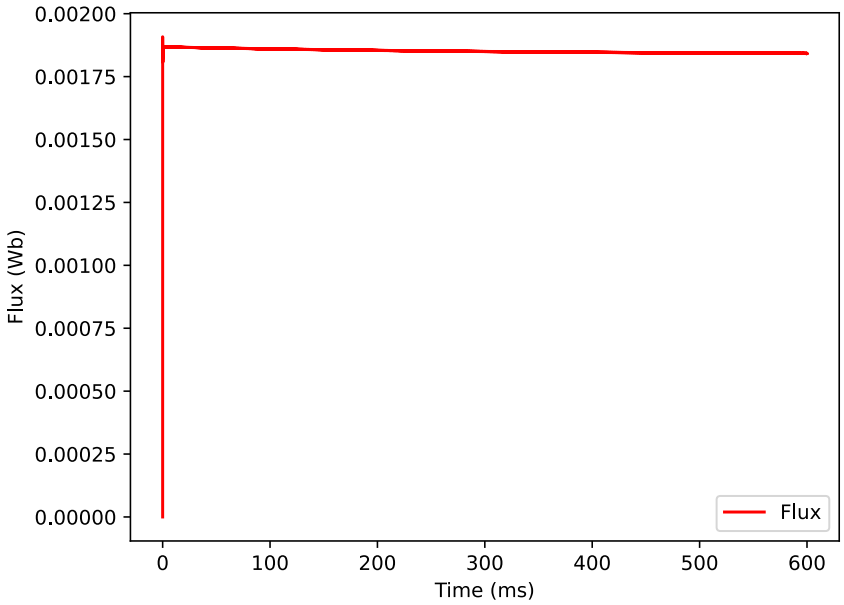

Figure I.108: Flux generated by the motor, for more information see code

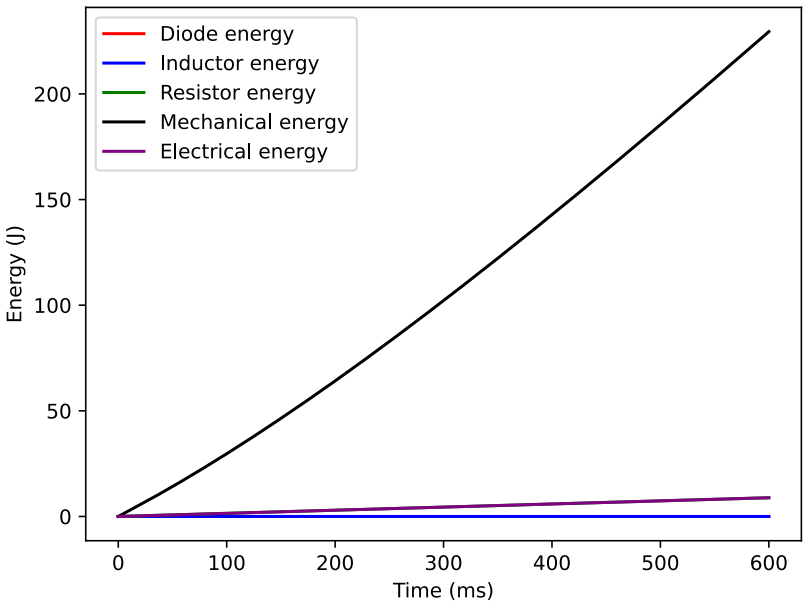

Figure I.109: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.963

900 rad/s

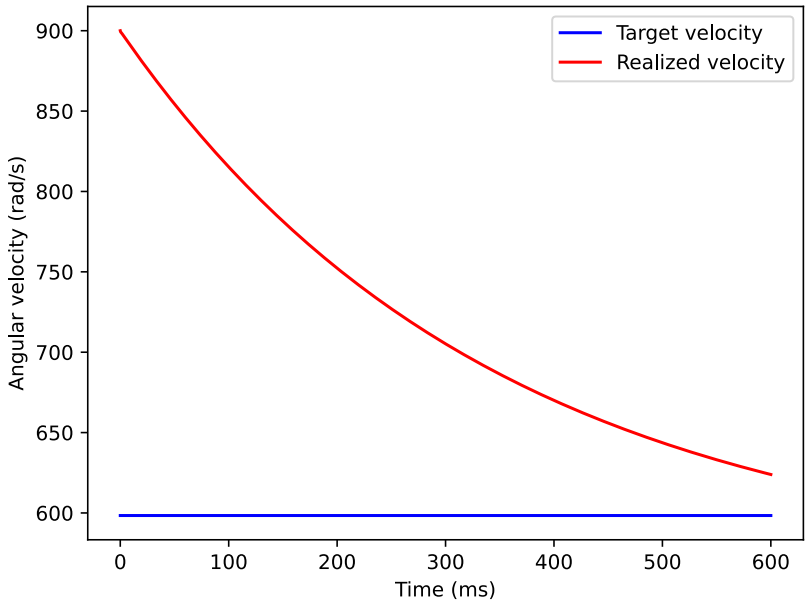

Figure I.110: The desired and realized velocity

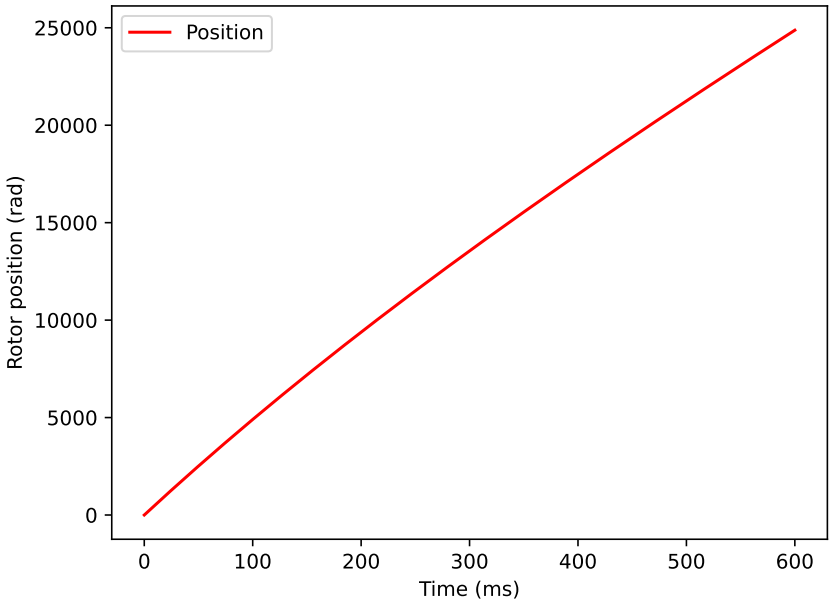

Figure I.111: The position of the rotor

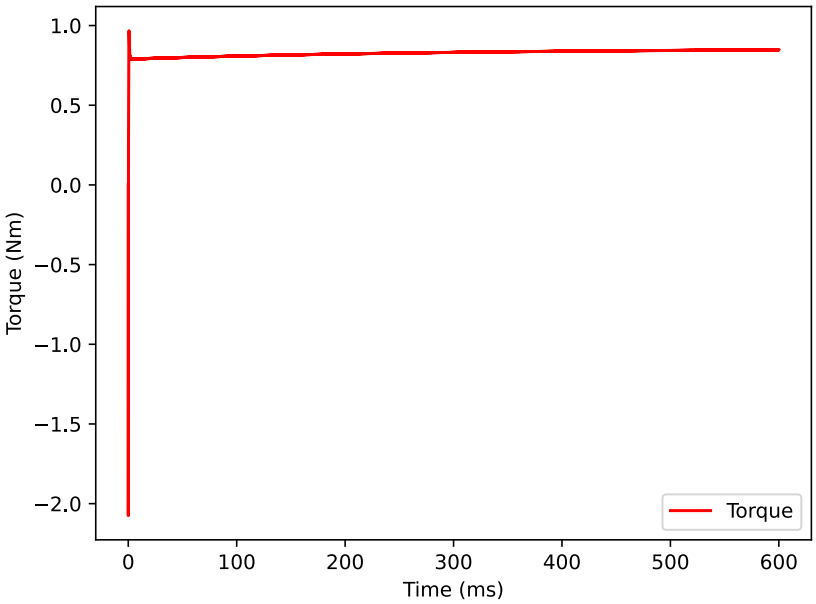

Figure I.112: The torque delivered by the motor

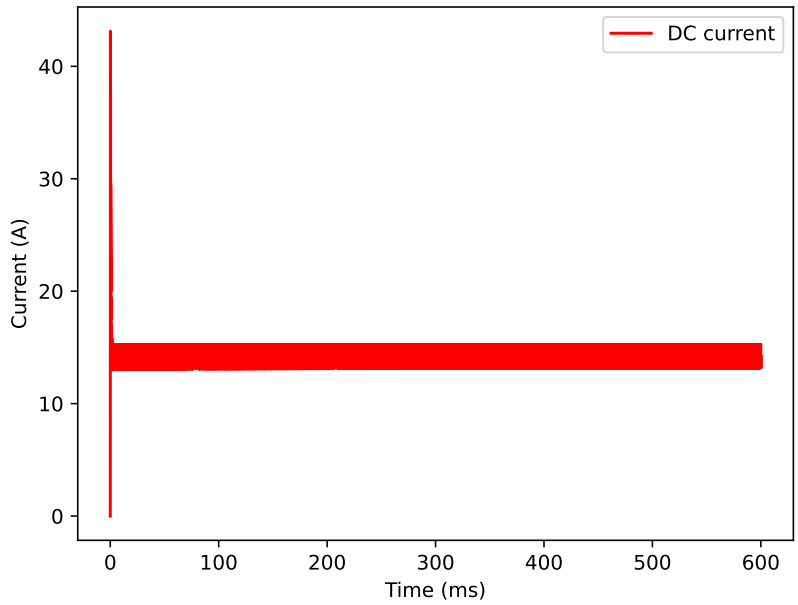

Figure I.113: The DC current through the motor

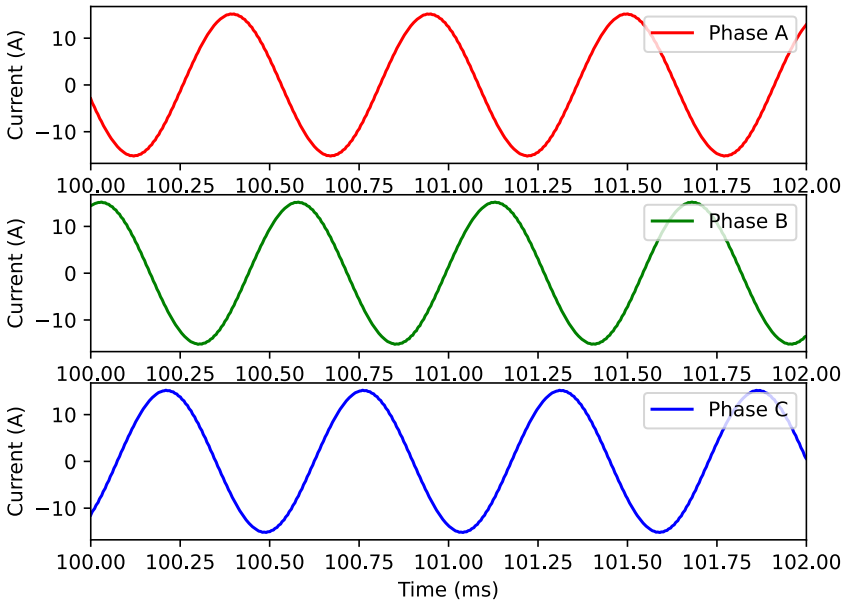

Figure I.114: Zoomed in phase currents through the motor

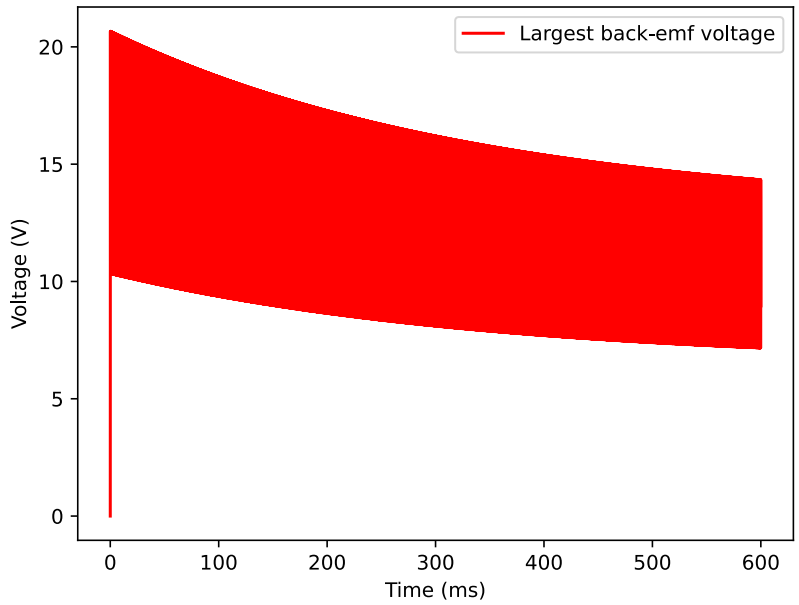

Figure I.115: The largest value for back-emf generated by a motor phase

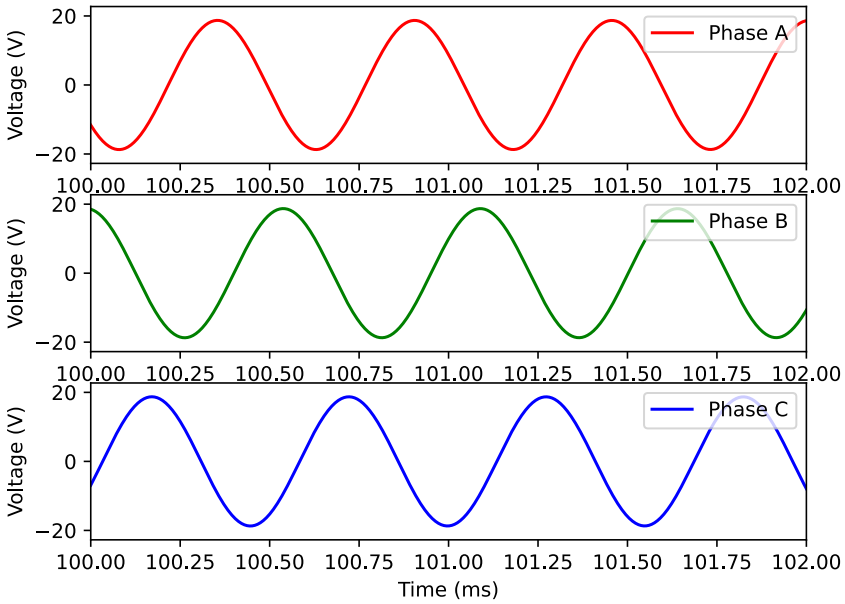

Figure I.116: Zoomed in back-emf generated by each phase

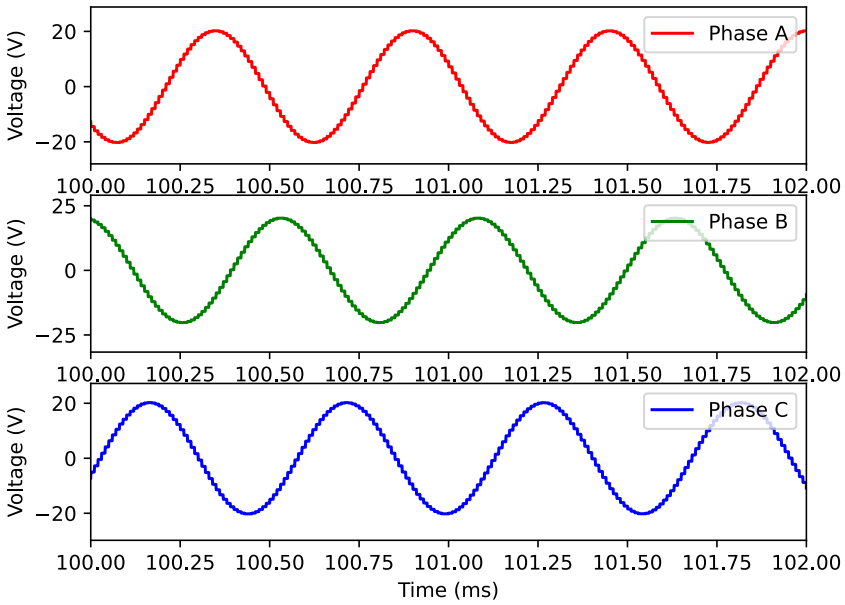

Figure I.117: Zoomed in phase voltages

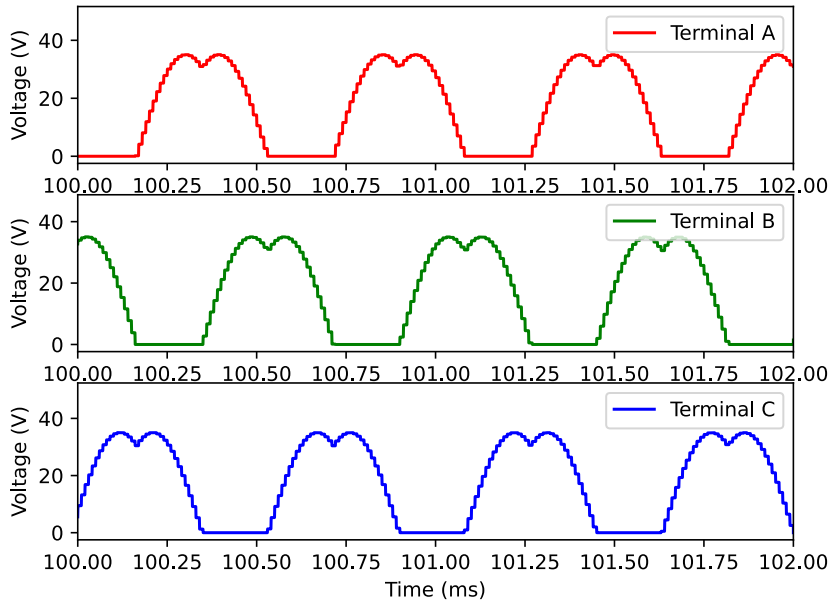

Figure I.118: Zoomed in terminal voltages

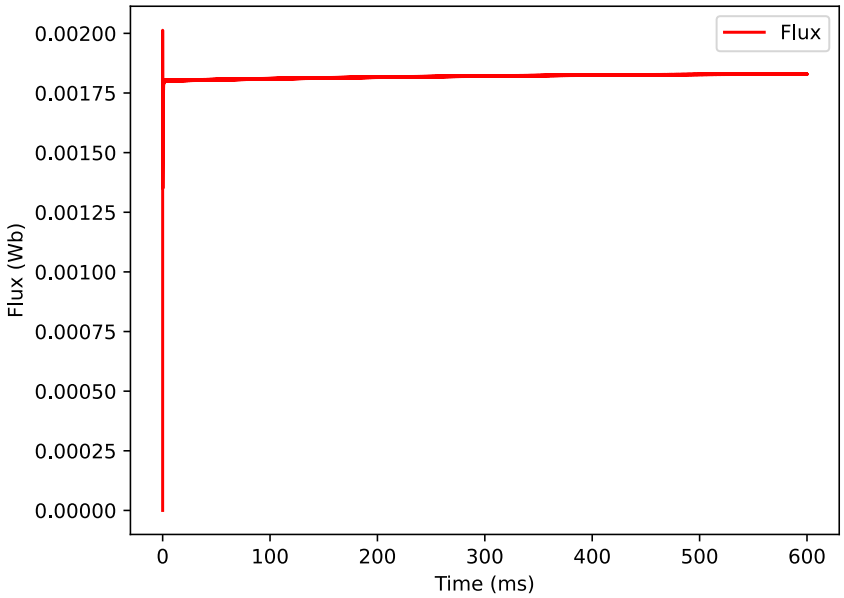

Figure I.119: Flux generated by the motor, for more information see code

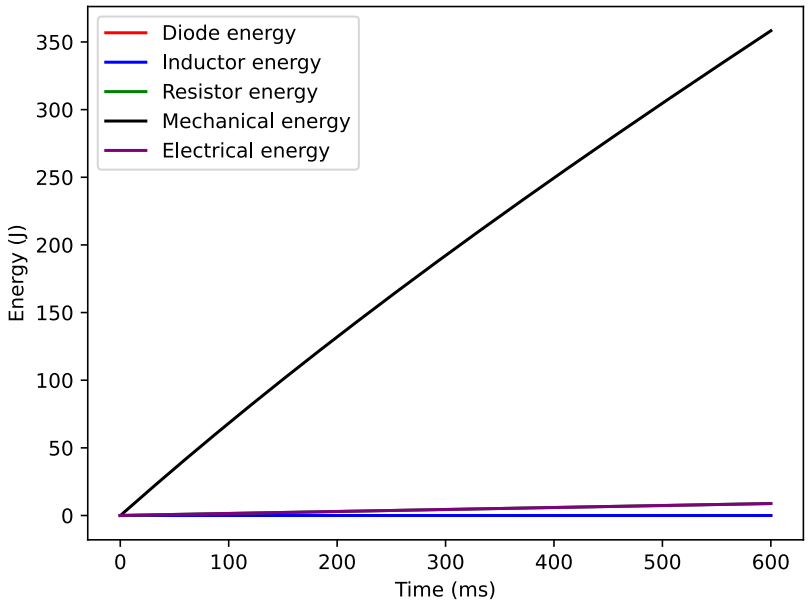

Figure I.120: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.976

Faulty measurements

Trapezoidal control

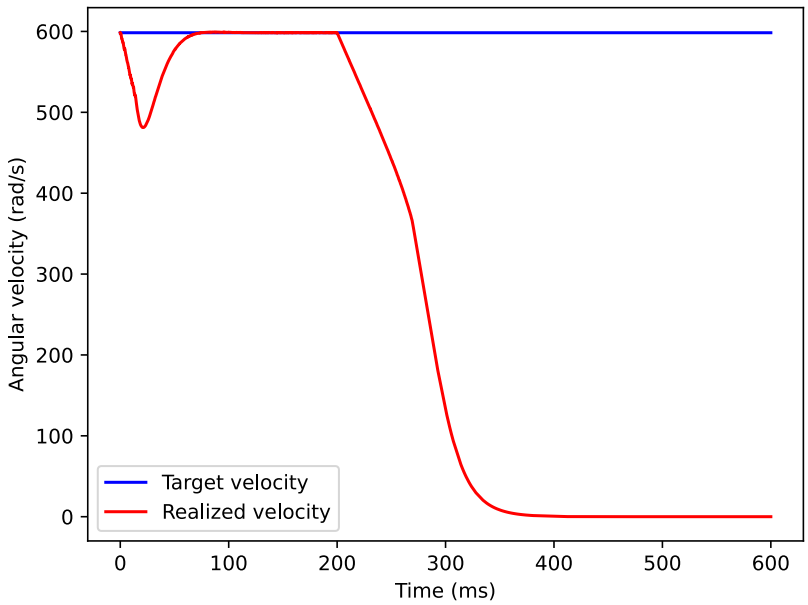

Figure I.121: The desired and realized velocity

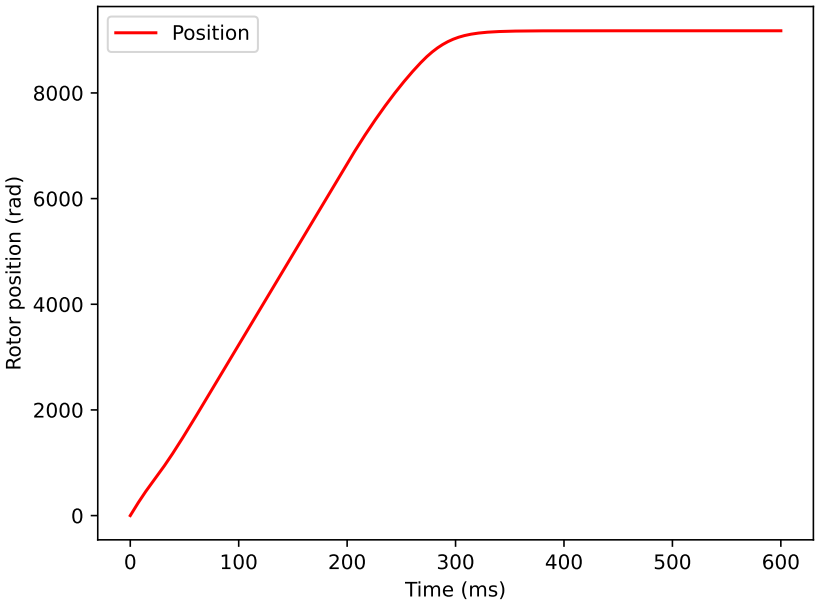

Figure I.122: The position of the rotor

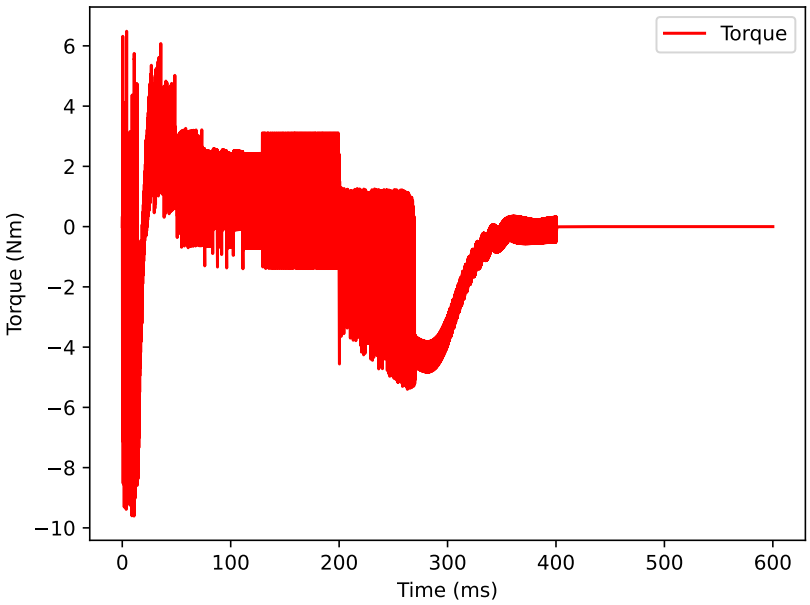

Figure I.123: The torque delivered by the motor

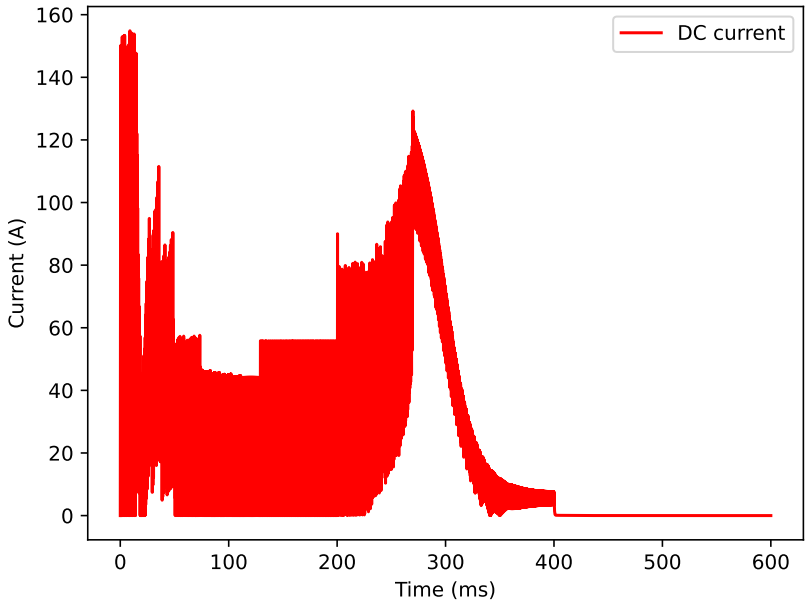

Figure I.124: The DC current through the motor

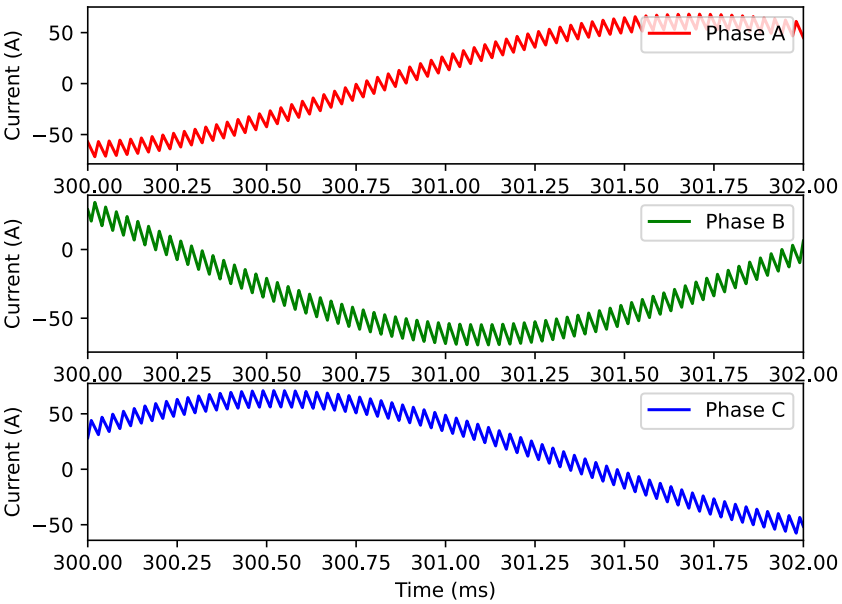

Figure I.125: Zoomed in phase currents through the motor

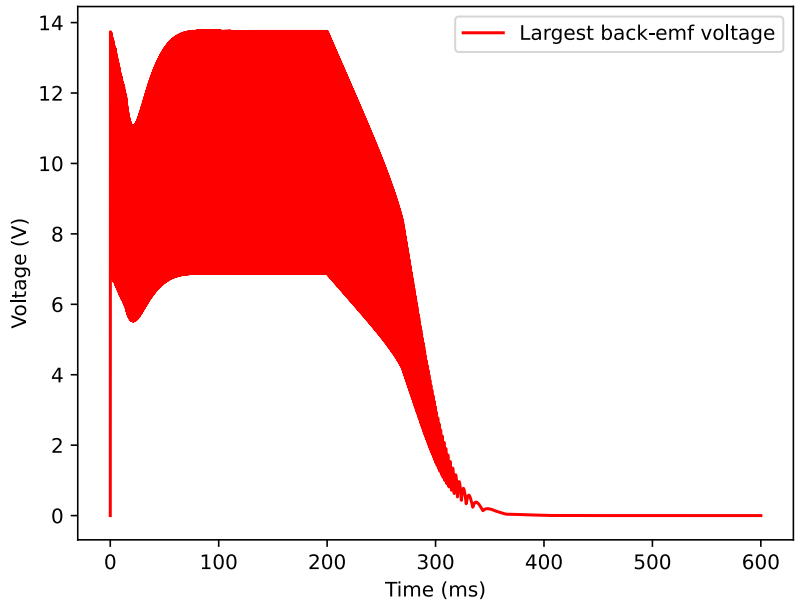

Figure I.126: The largest value for back-emf generated by a motor phase

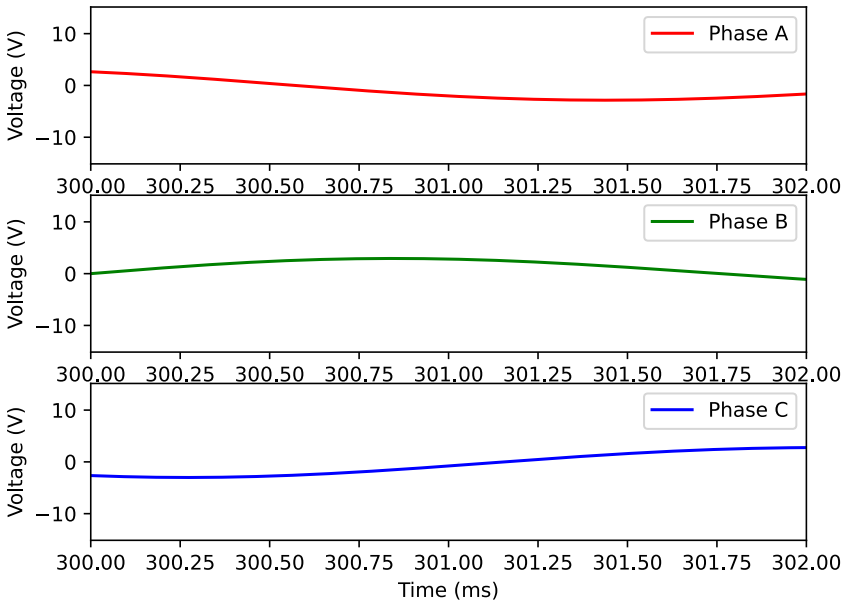

Figure I.127: Zoomed in back-emf generated by each phase

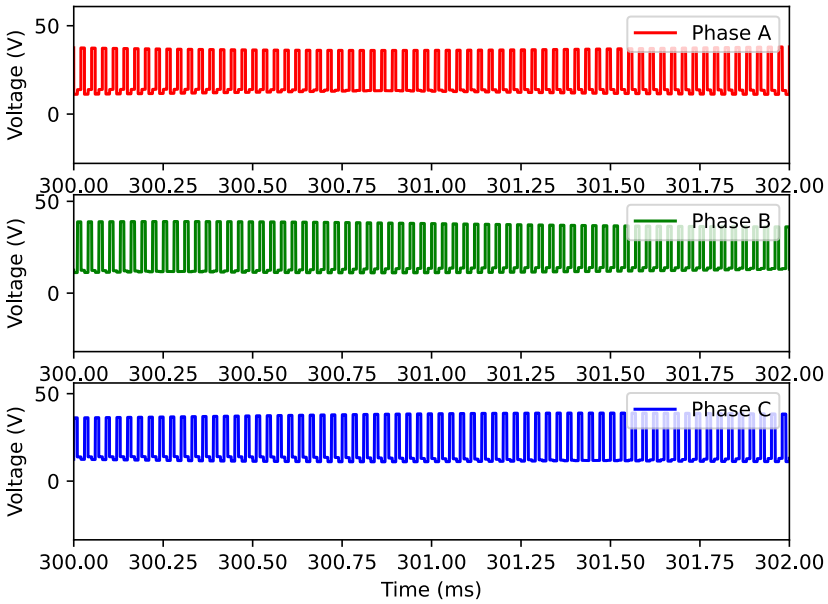

Figure I.128: Zoomed in phase voltages

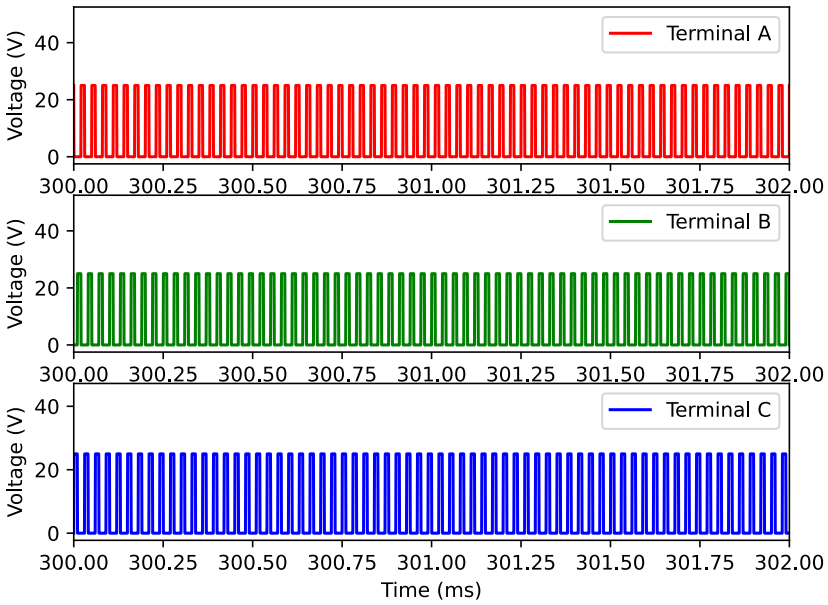

Figure I.129: Zoomed in terminal voltages

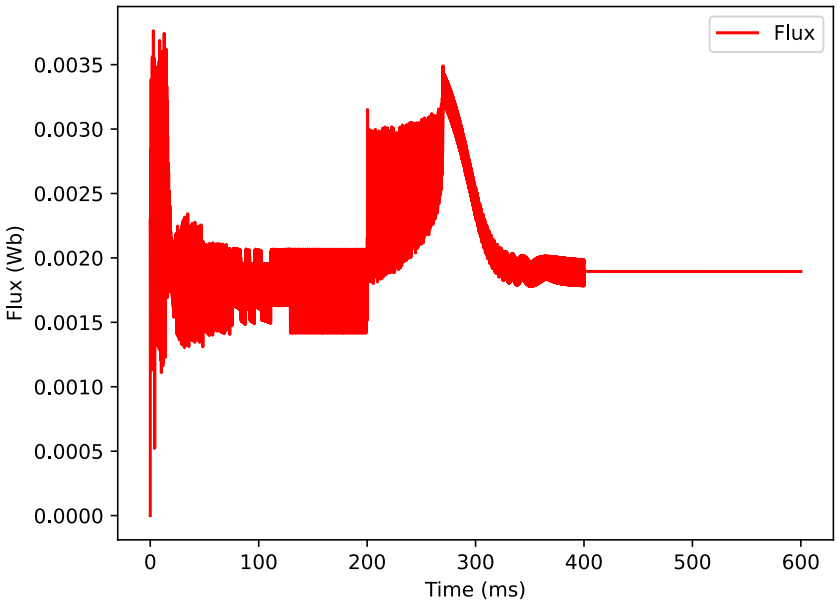

Figure I.130: Flux generated by the motor, for more information see code

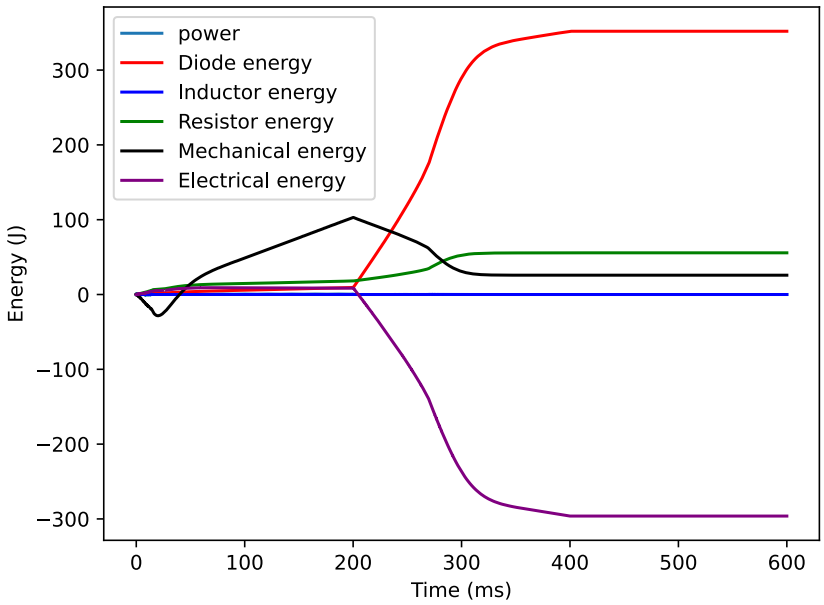

Figure I.131: The energy used or generated by various components, the total electrical energy and the total mechanical energy.

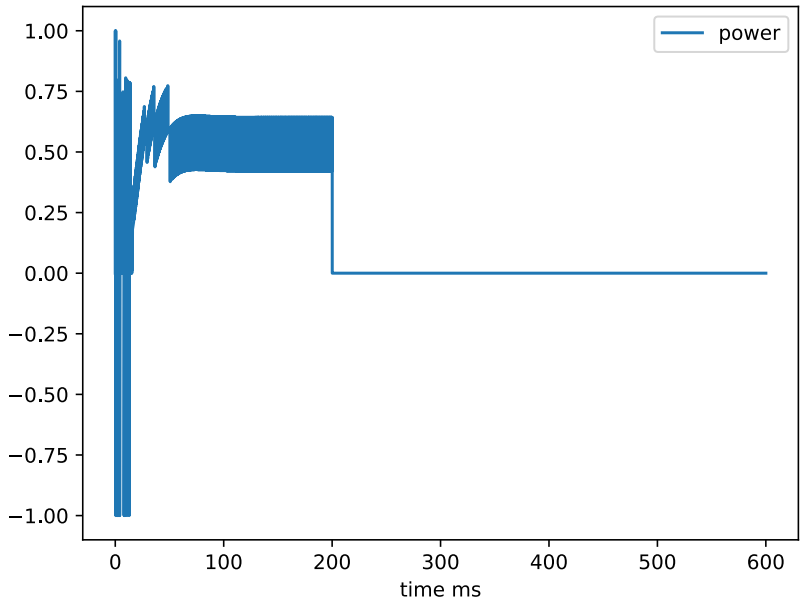

Figure I.132: The amplitude of the normalized output voltage

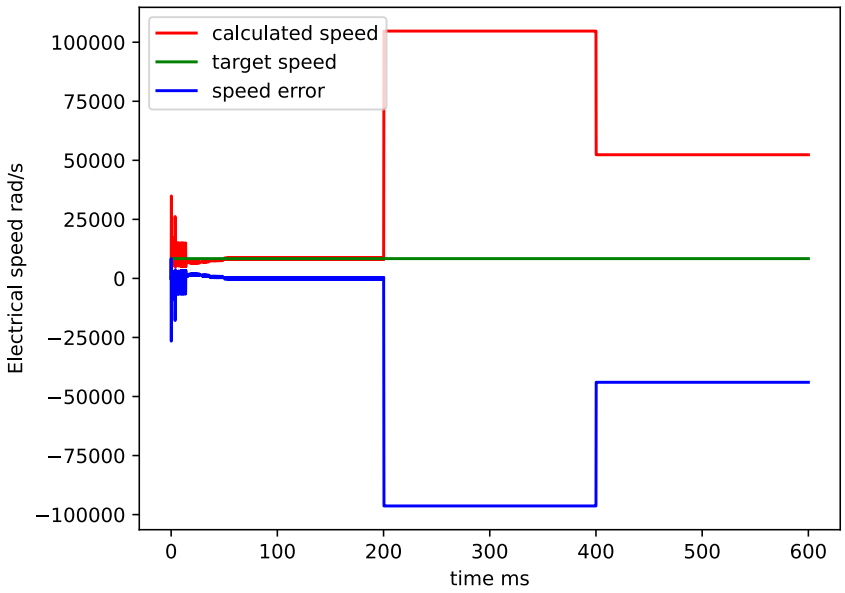

Figure I.133: The calculated, target, and error in the electrical speed of the trapezoidal controller

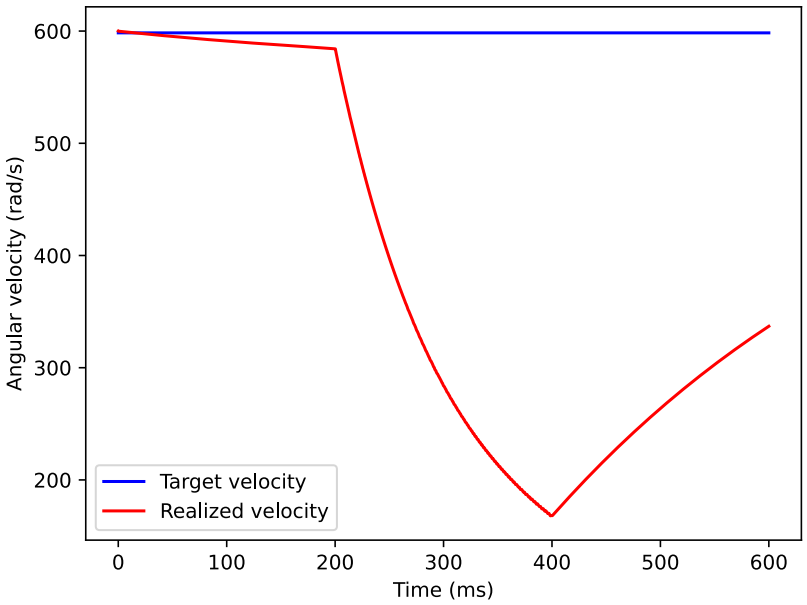

Figure I.134: The desired and realized velocity

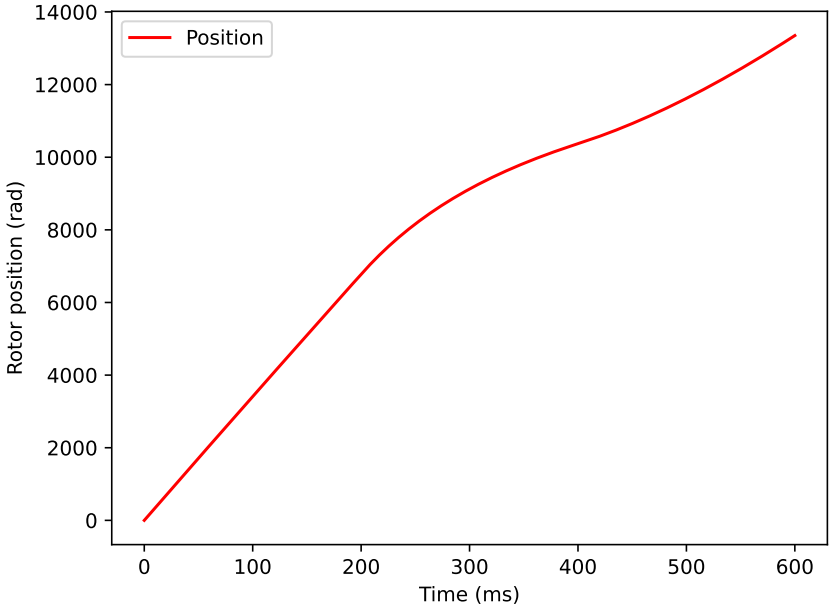

Figure I.135: The position of the rotor

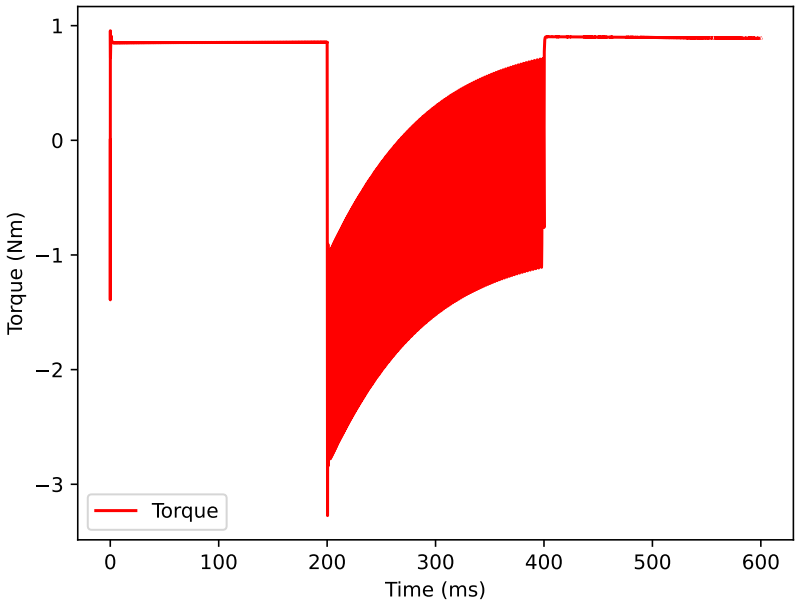

Figure I.136: The torque delivered by the motor

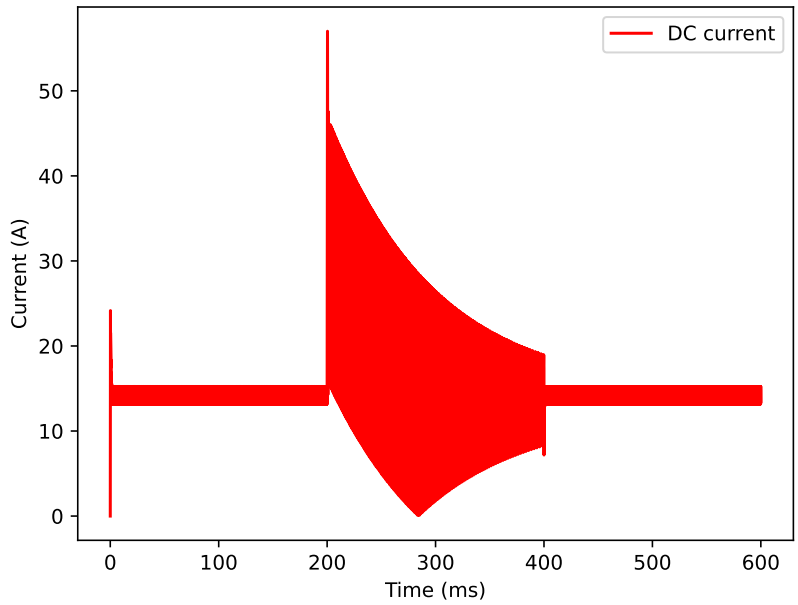

Figure I.137: The DC current through the motor

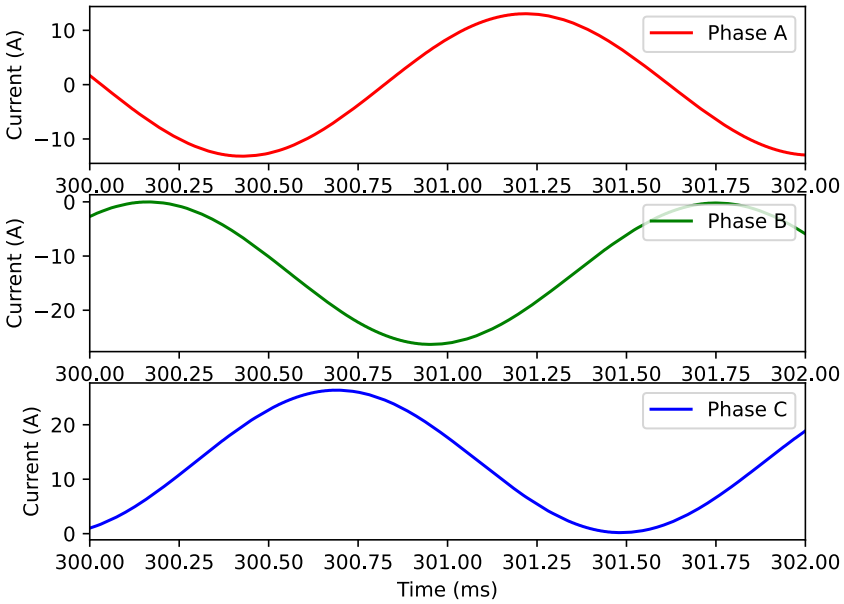

Figure I.138: Zoomed in phase currents through the motor

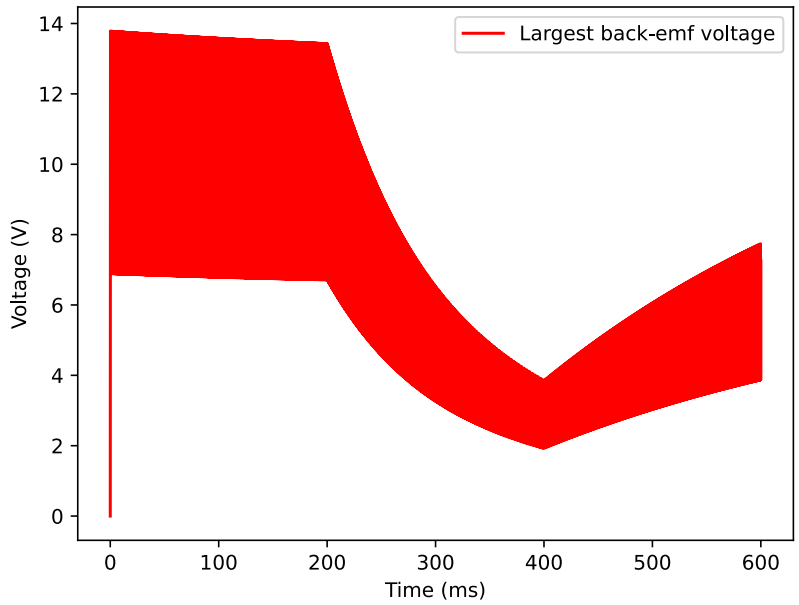

Figure I.139: The largest value for back-emf generated by a motor phase

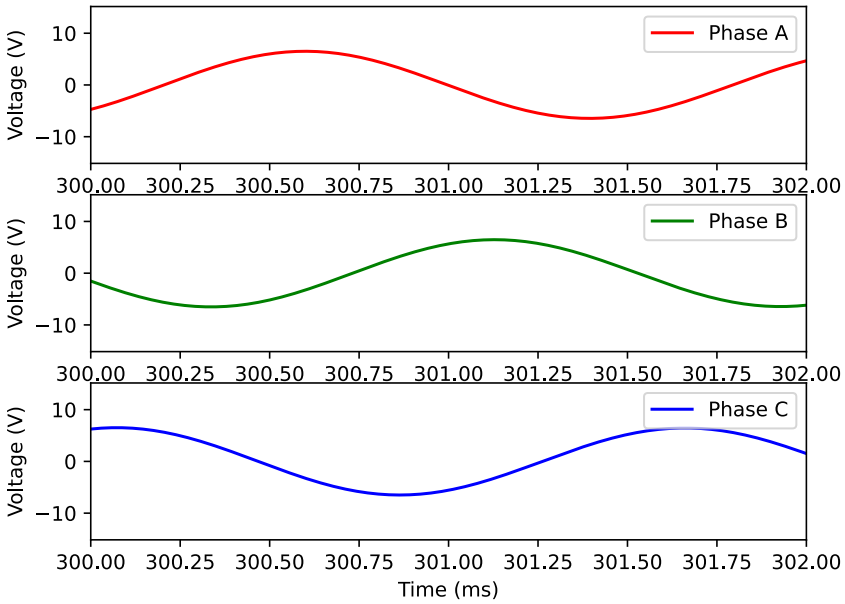

Figure I.140: Zoomed in back-emf generated by each phase

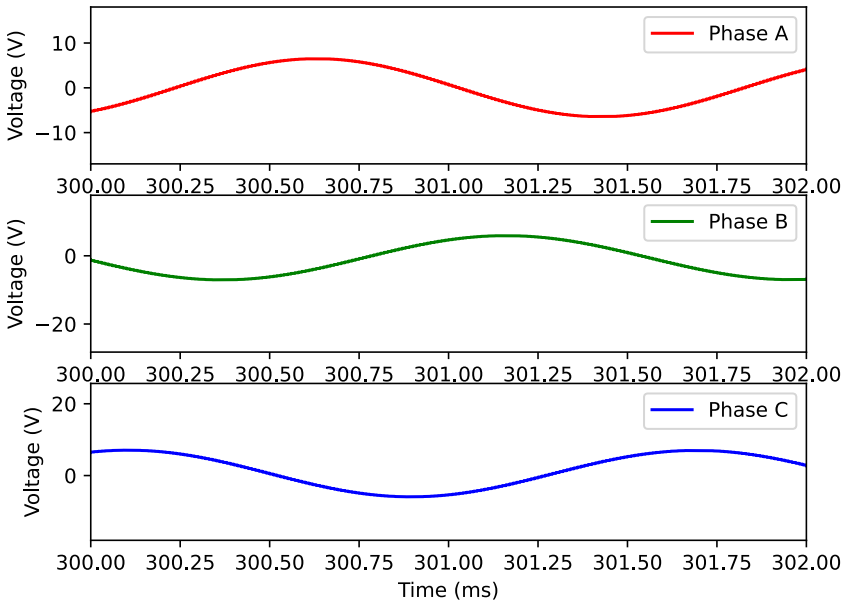

Figure I.141: Zoomed in phase voltages

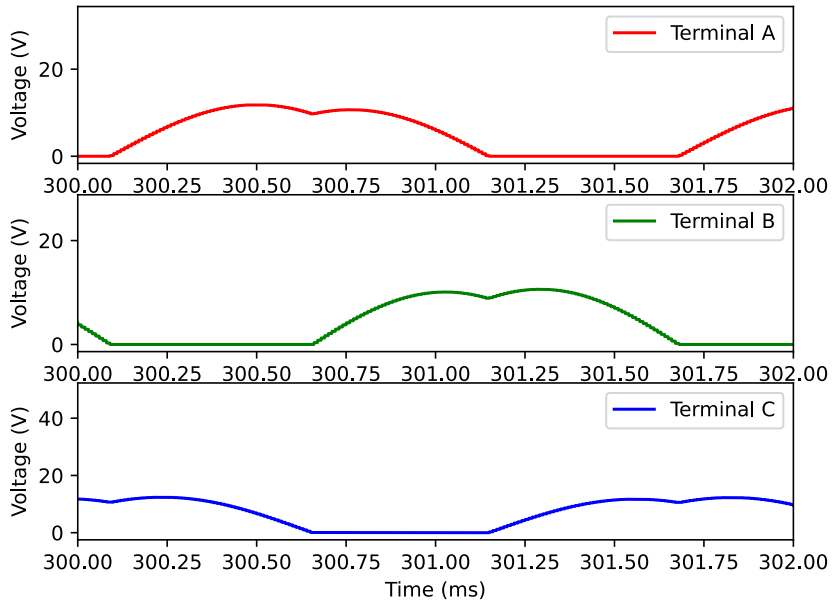

Figure I.142: Zoomed in terminal voltages

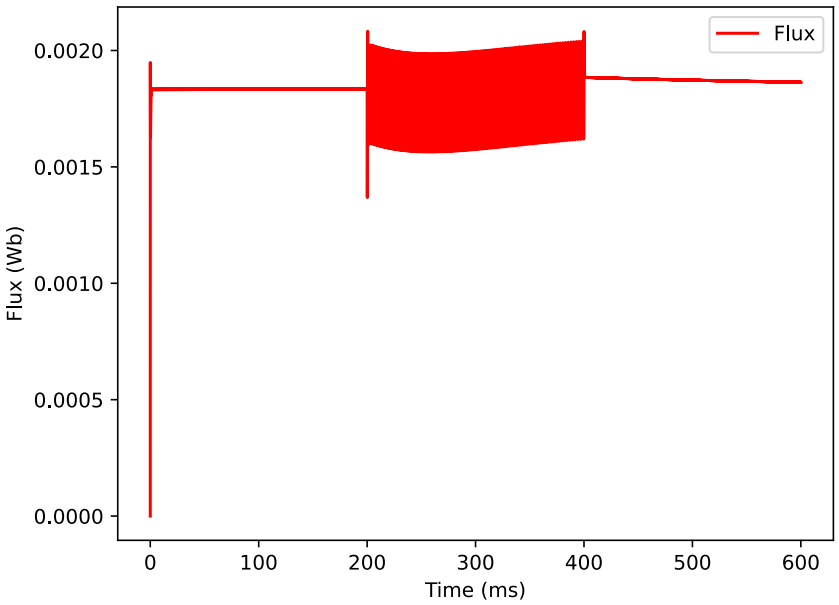

Figure I.143: Flux generated by the motor, for more information see code

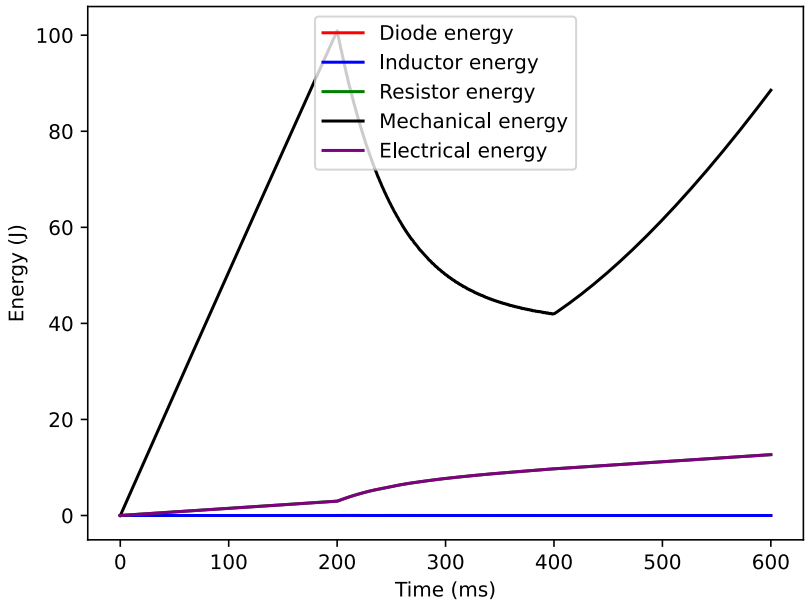

Figure I.144: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.875

Sudden change in velocity

Trapezoidal control

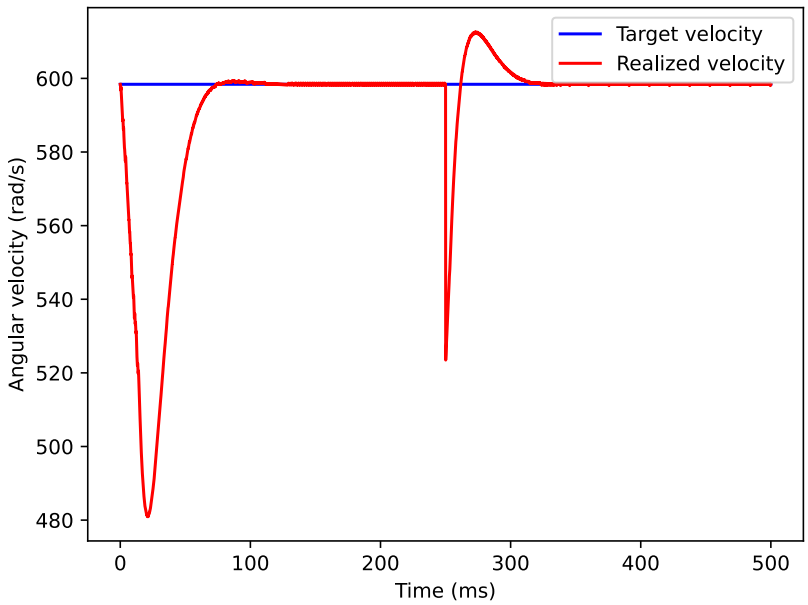

Figure I.145: The desired and realized velocity

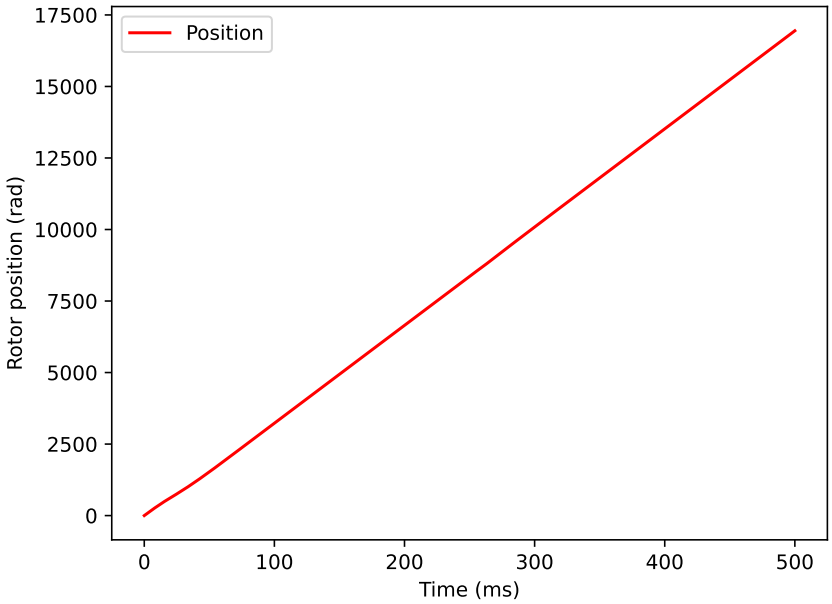

Figure I.146: The position of the rotor

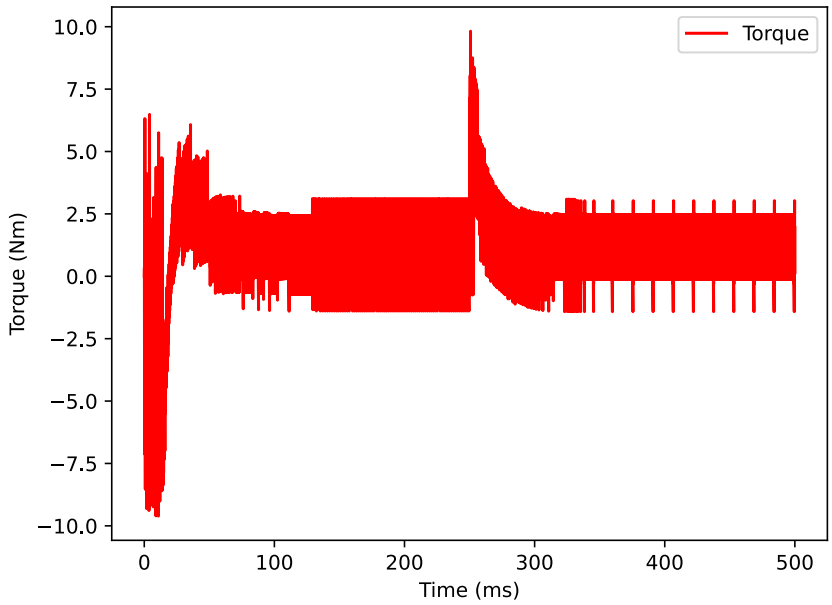

Figure I.147: The torque delivered by the motor

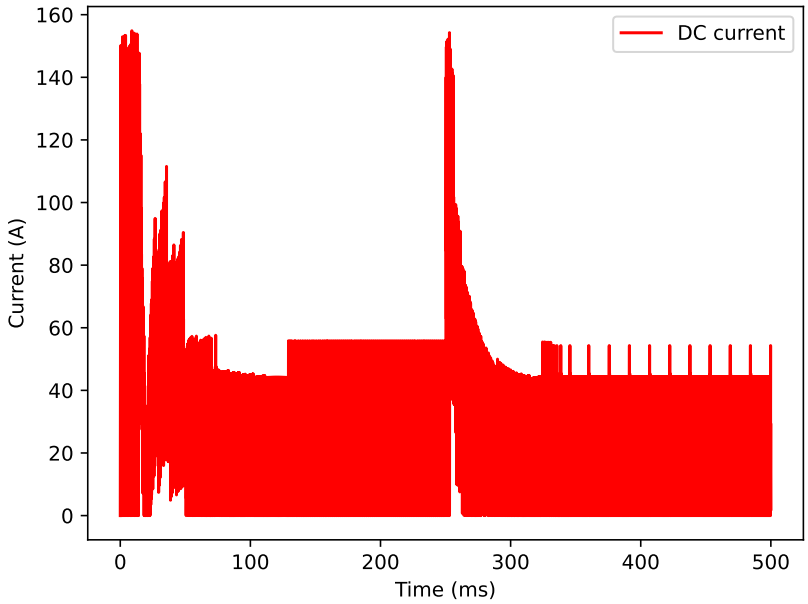

Figure I.148: The DC current through the motor

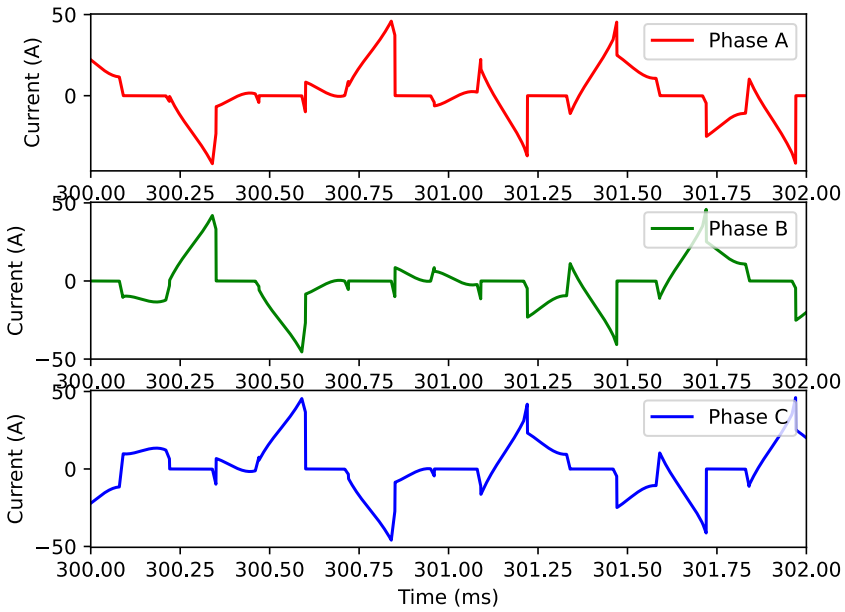

Figure I.149: Zoomed in phase currents through the motor

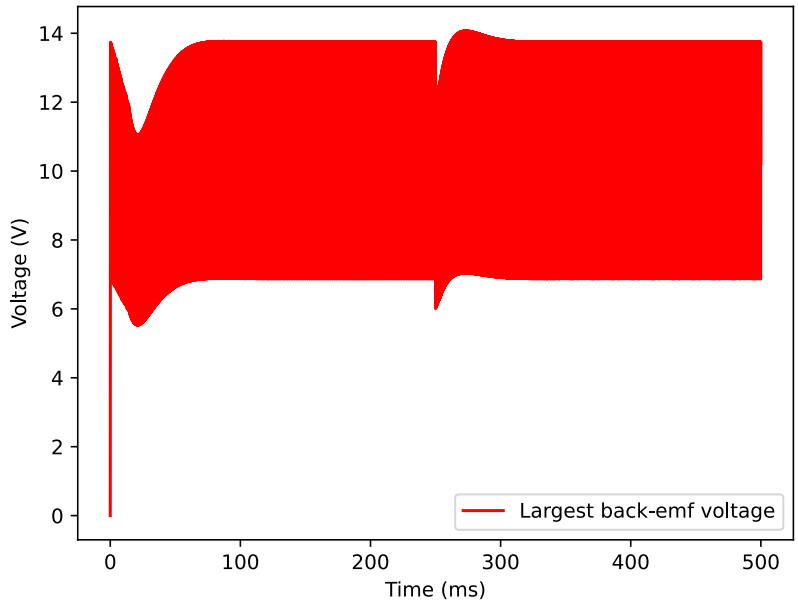

Figure I.150: The largest value for back-emf generated by a motor phase

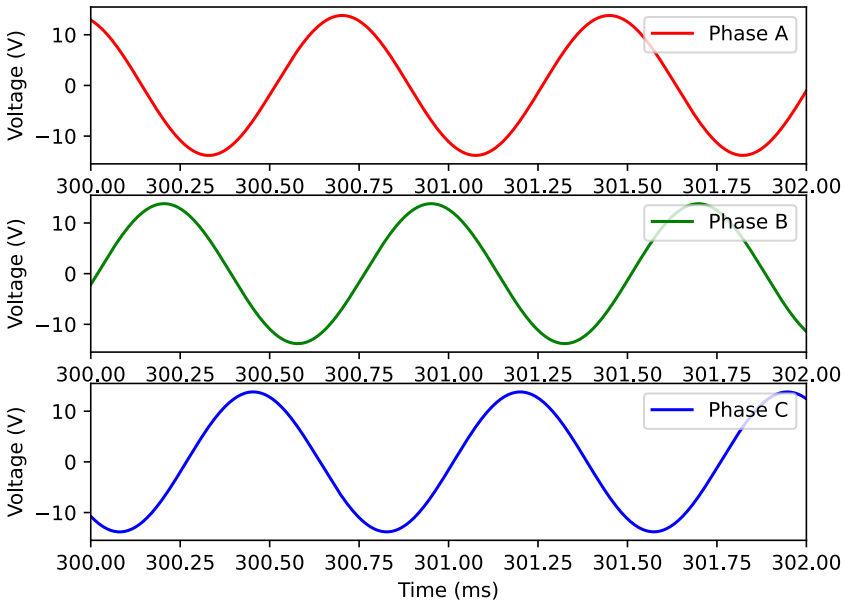

Figure I.151: Zoomed in back-emf generated by each phase

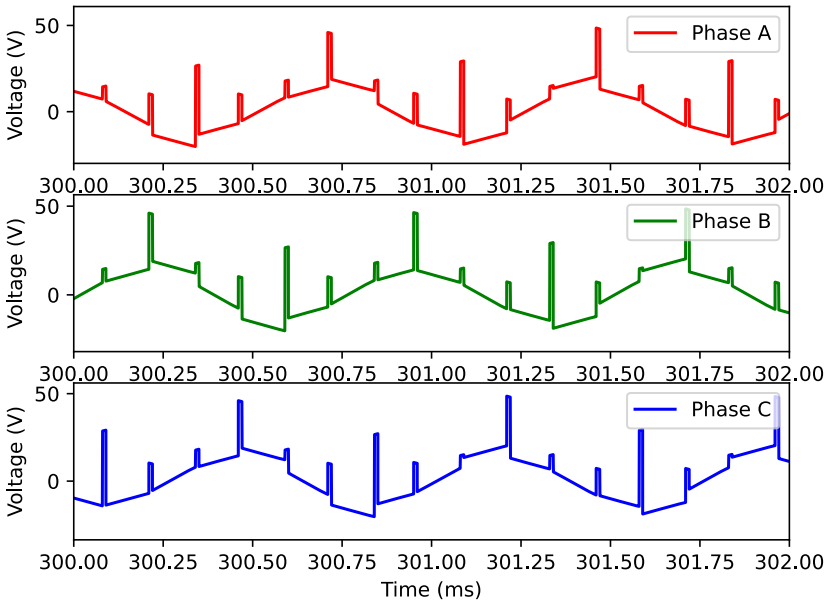

Figure I.152: Zoomed in phase voltages

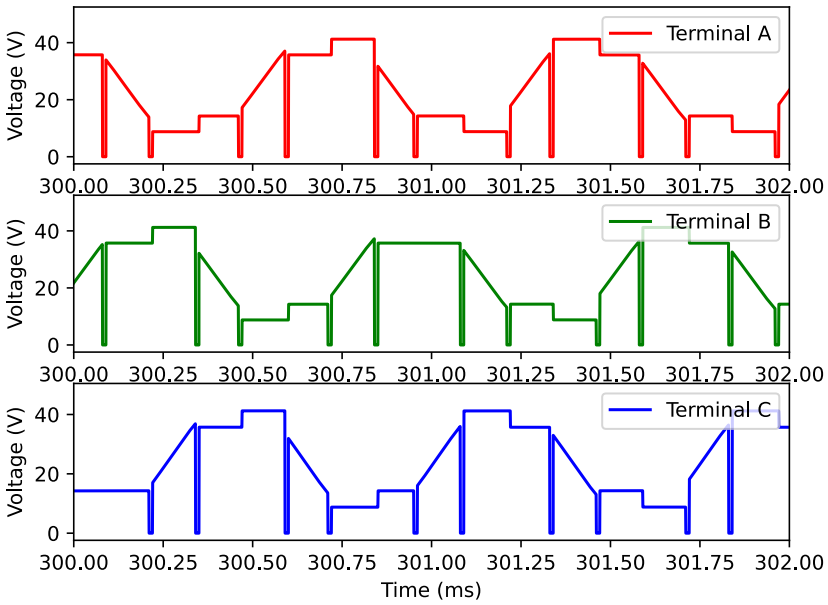

Figure I.153: Zoomed in terminal voltages

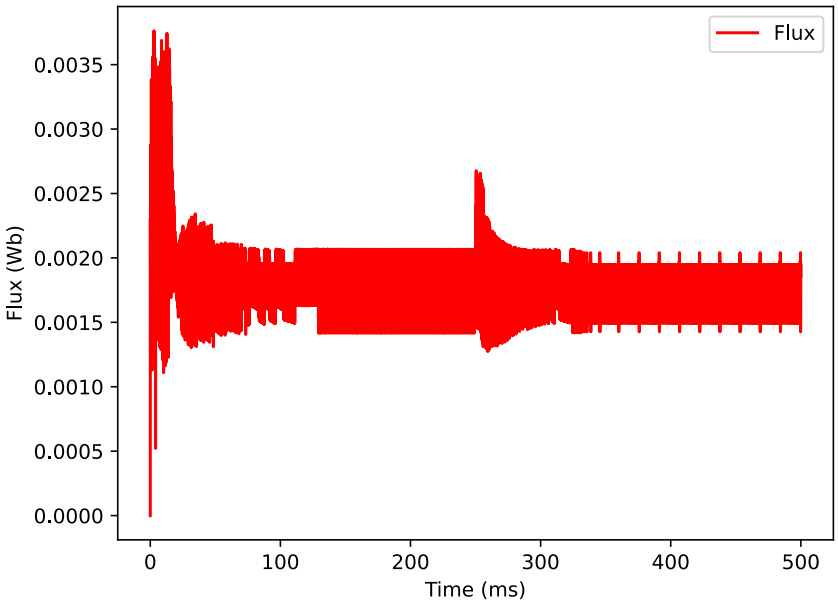

Figure I.154: Flux generated by the motor, for more information see code

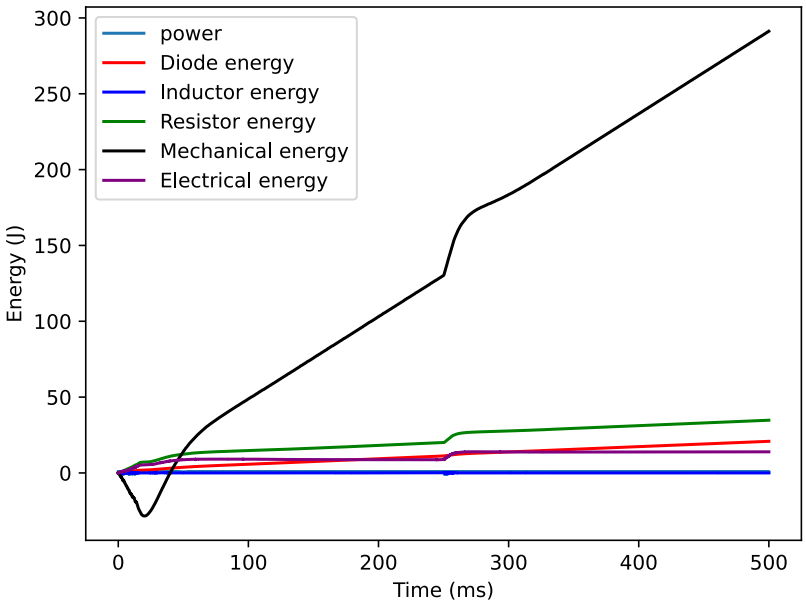

Figure I.155: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.954

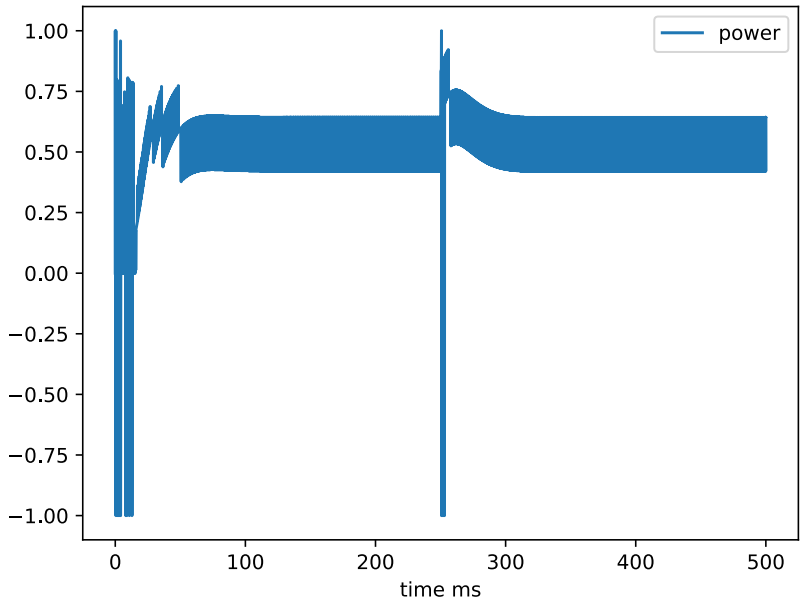

Figure I.156: The amplitude of the normalized output voltage

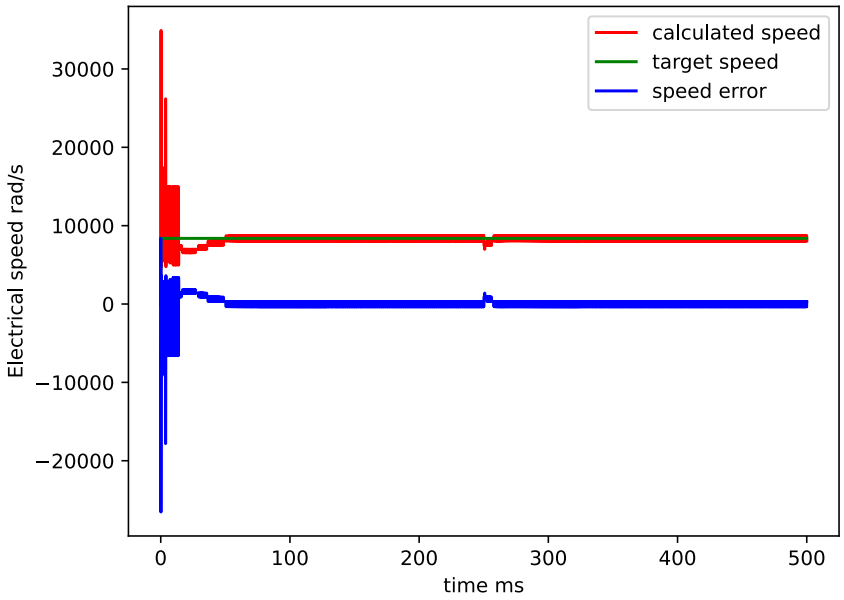

Figure I.157: The calculated, target, and error in the electrical speed of the trapezoidal controller

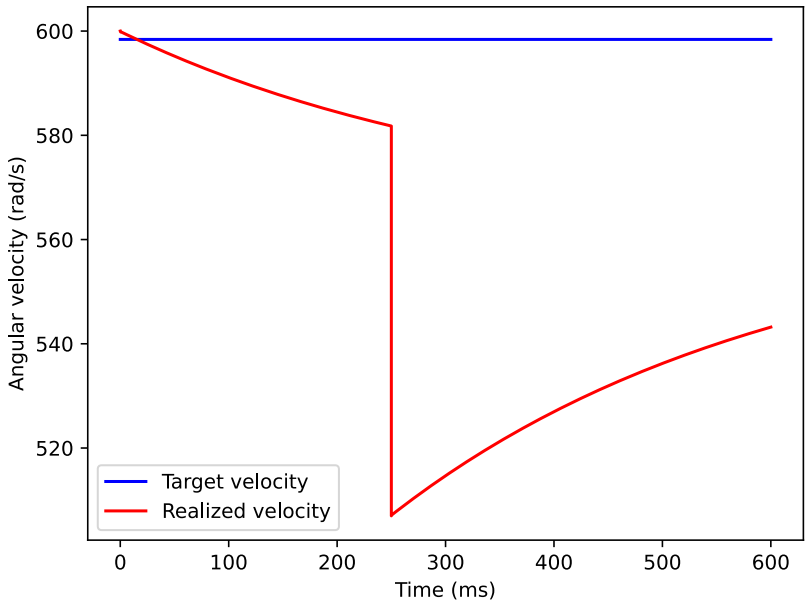

Figure I.158: The desired and realized velocity

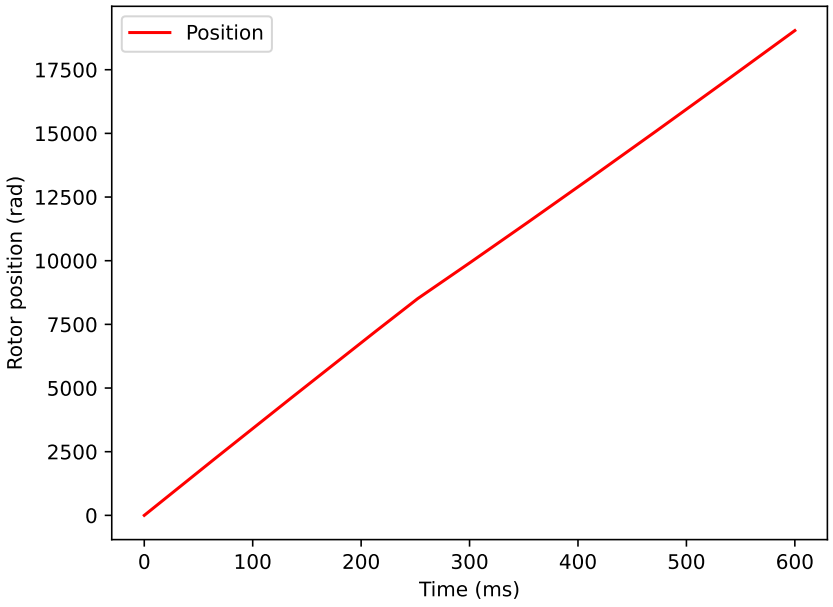

Figure I.159: The position of the rotor

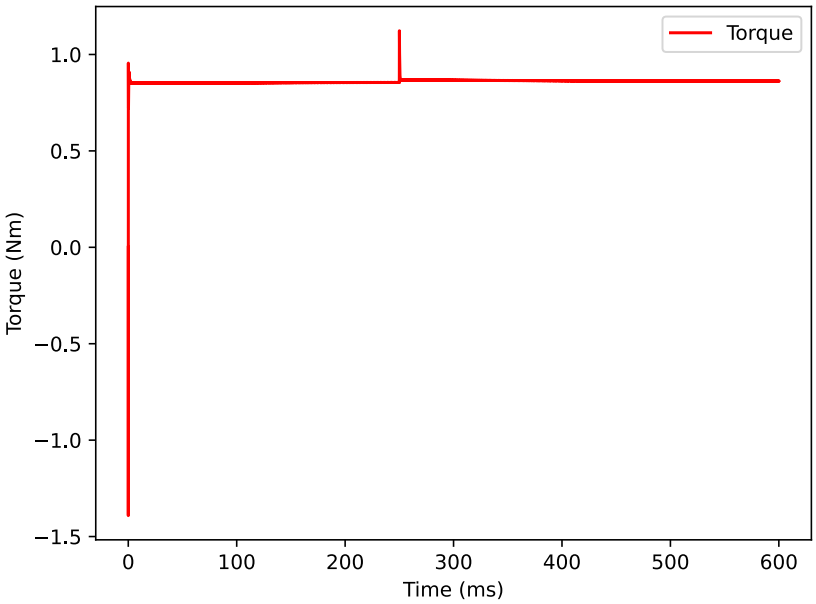

Figure I.160: The torque delivered by the motor

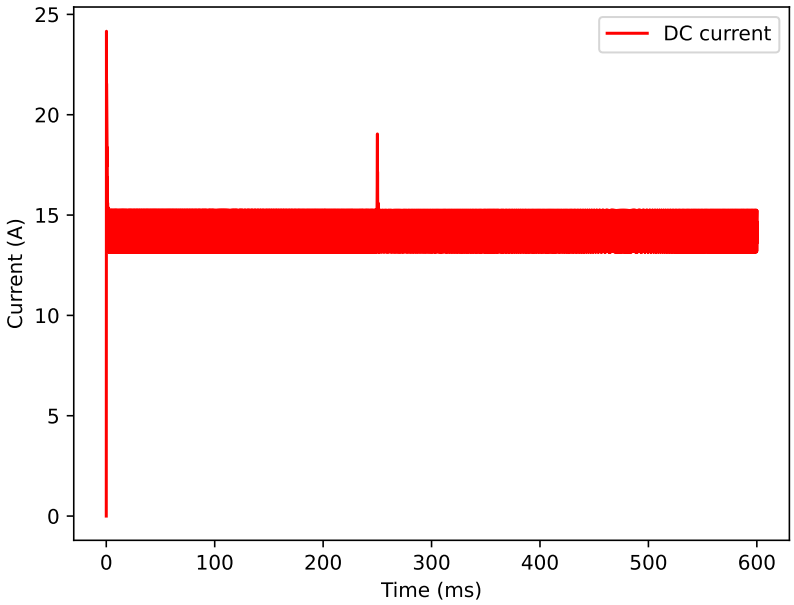

Figure I.161: The DC current through the motor

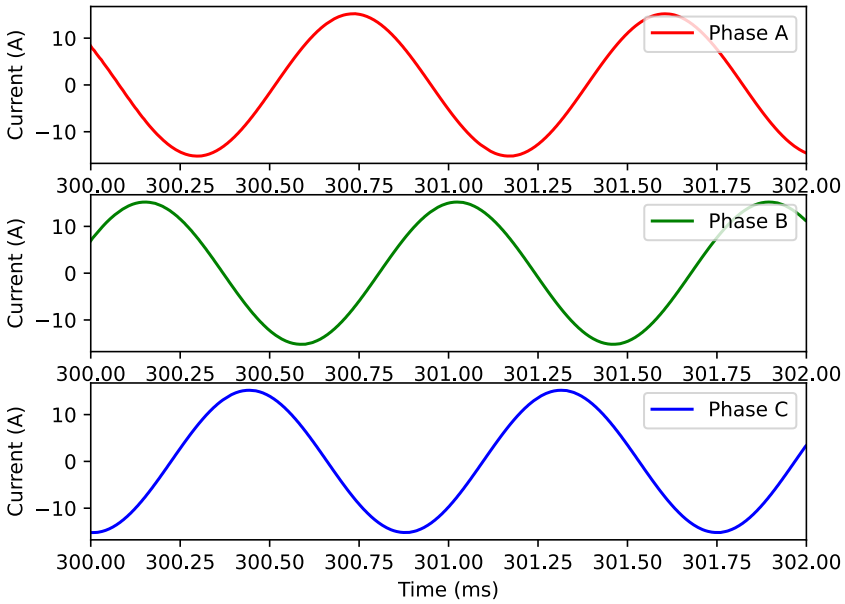

Figure I.162: Zoomed in phase currents through the motor

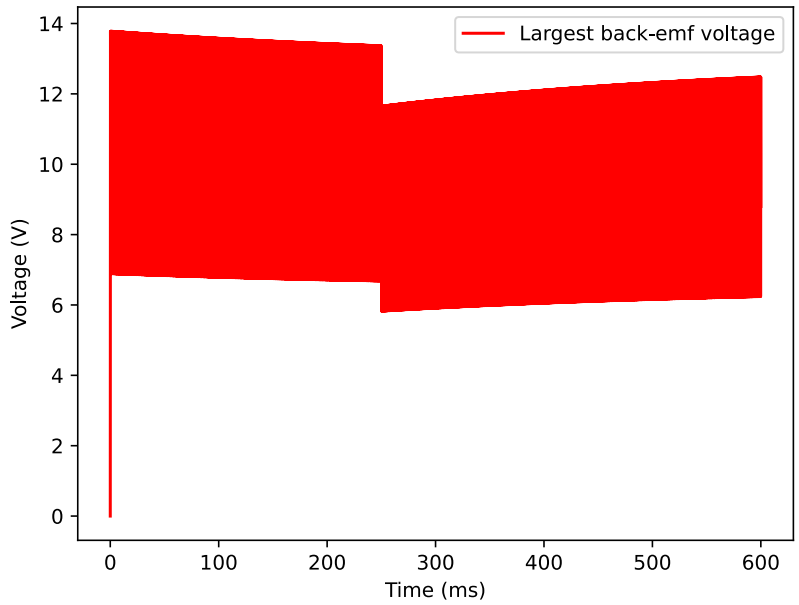

Figure I.163: The largest value for back-emf generated by a motor phase

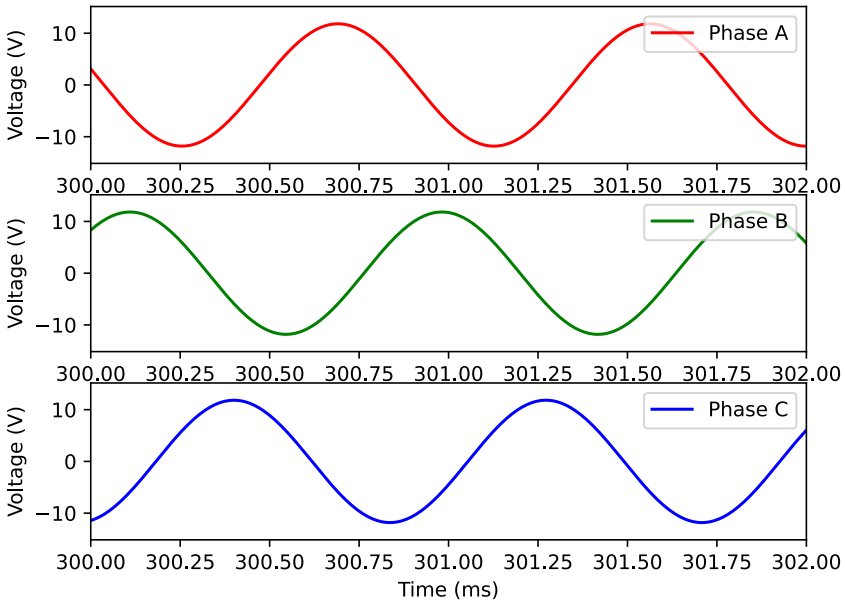

Figure I.164: Zoomed in back-emf generated by each phase

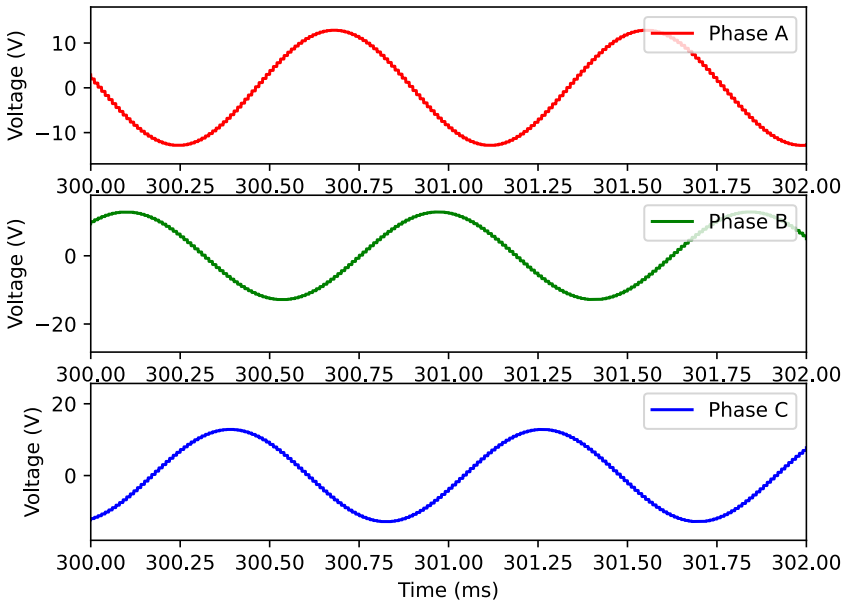

Figure I.165: Zoomed in phase voltages

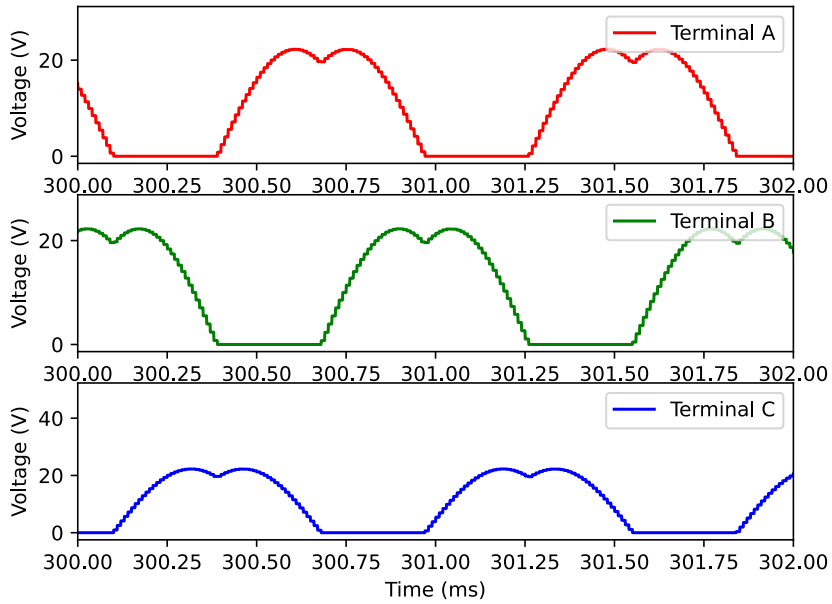

Figure I.166: Zoomed in terminal voltages

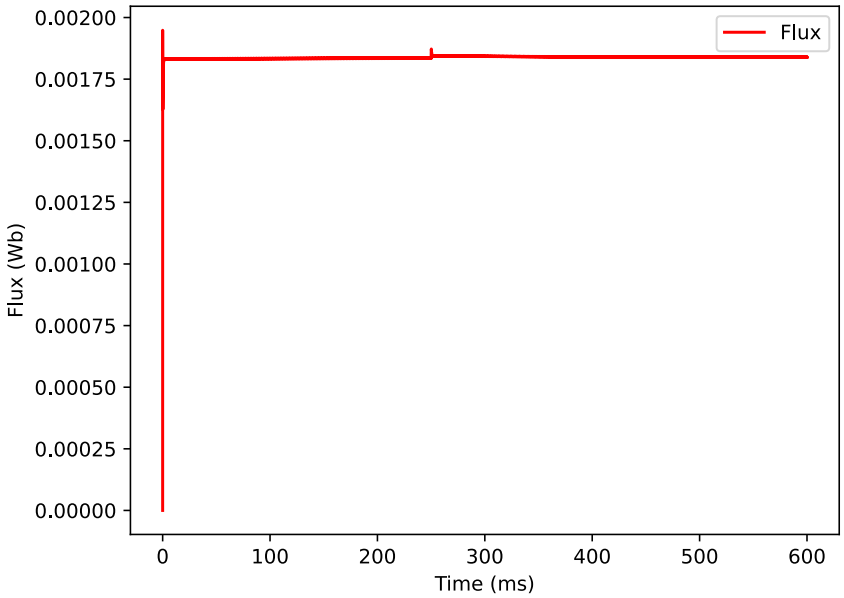

Figure I.167: Flux generated by the motor, for more information see code

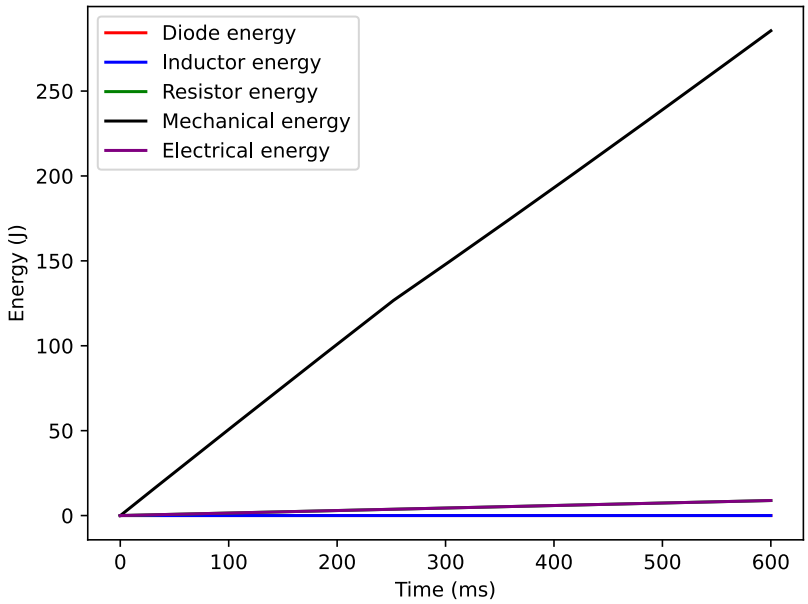

Figure I.168: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.970

Efficiency in steady state

Trapezoidal control

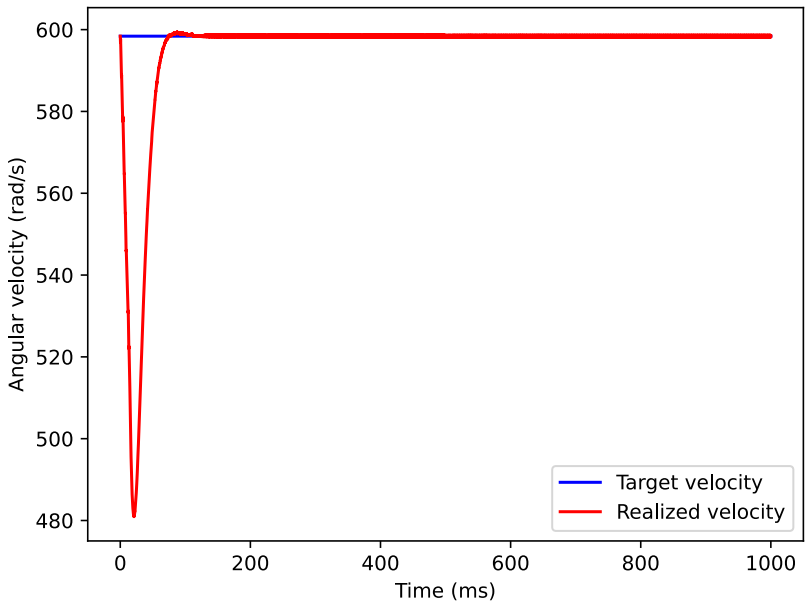

Figure I.169: The desired and realized velocity

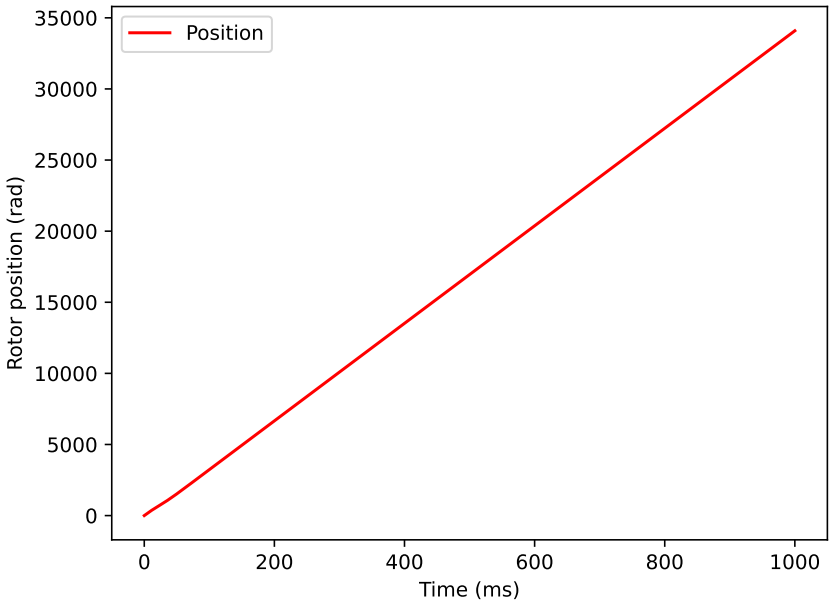

Figure I.170: The position of the rotor

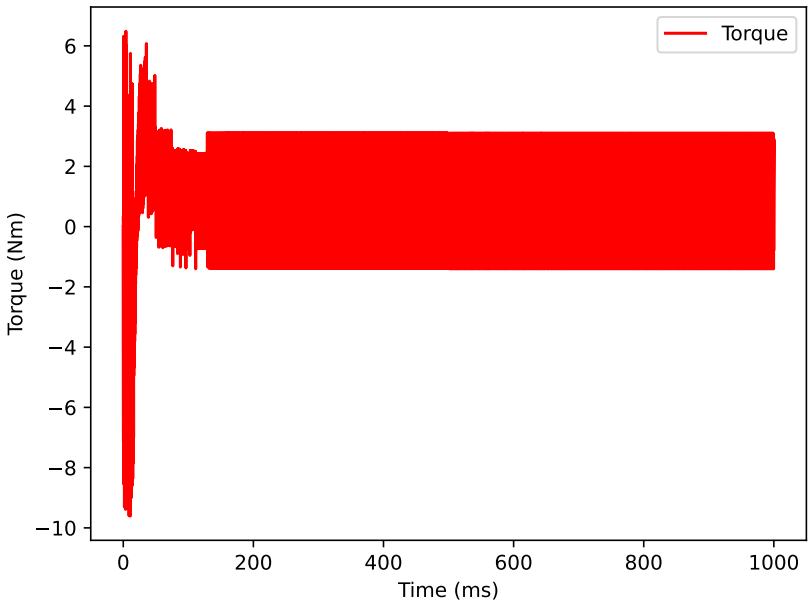

Figure I.171: The torque delivered by the motor

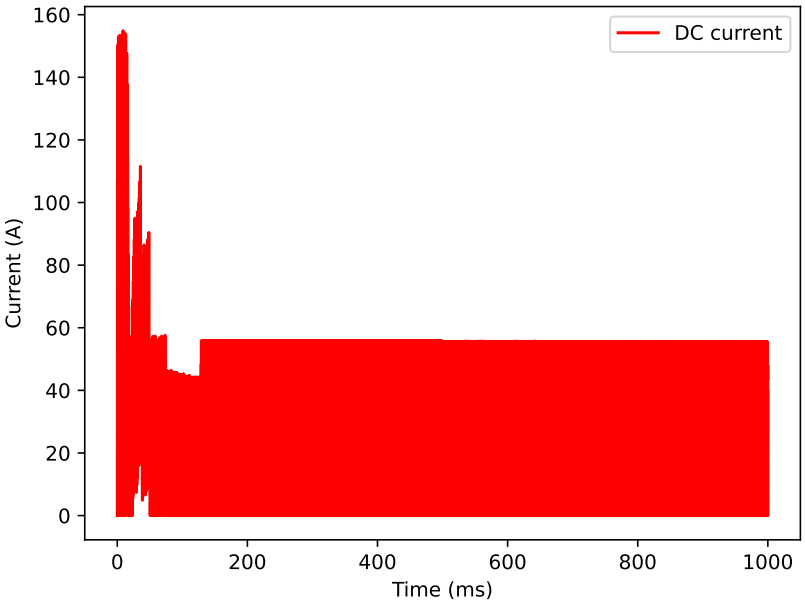

Figure I.172: The DC current through the motor

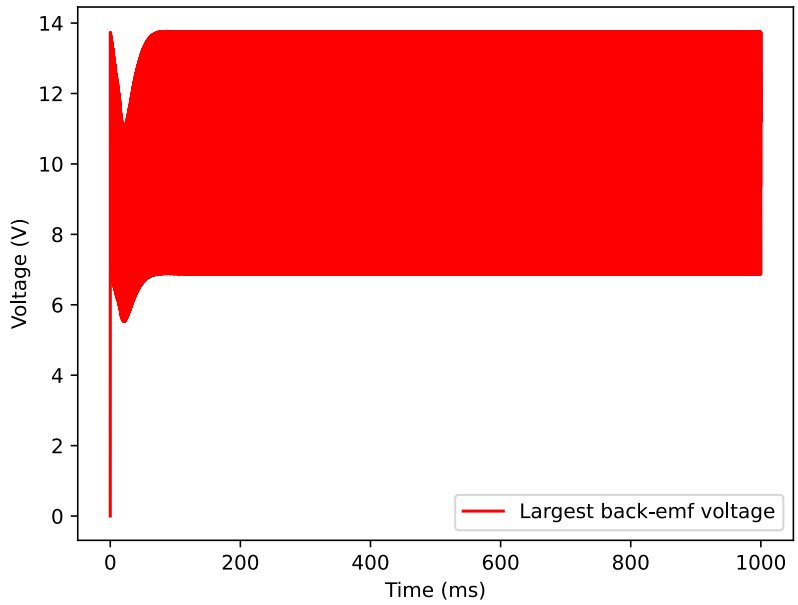

Figure I.173: The largest value for back-emf generated by a motor phase

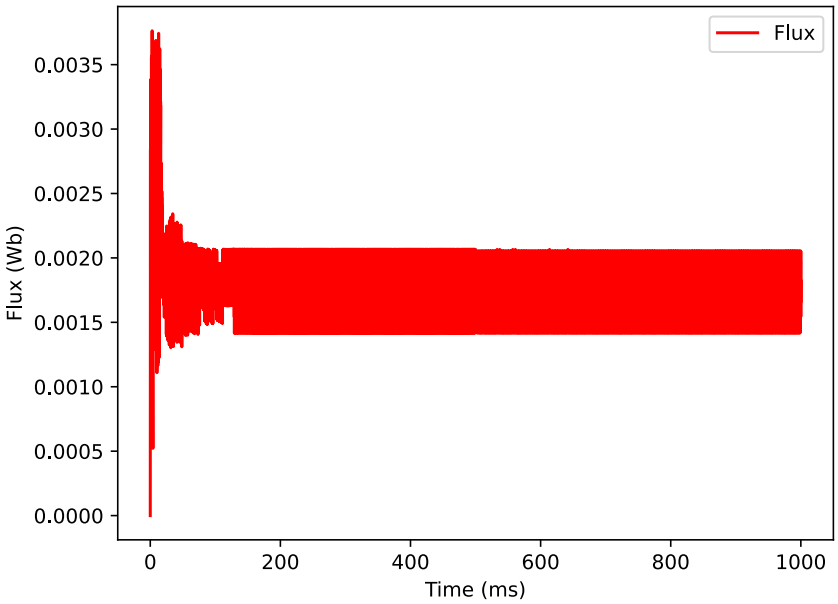

Figure I.174: Flux generated by the motor, for more information see code

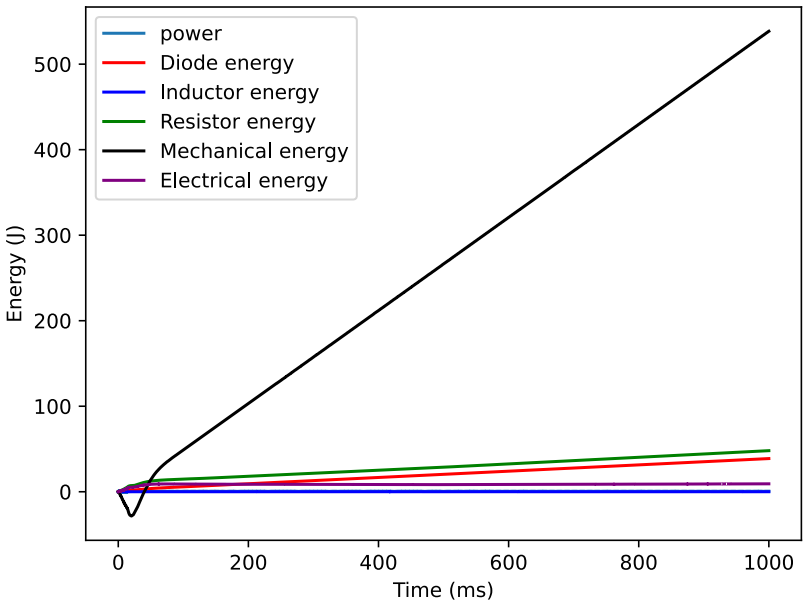

Figure I.175: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.983

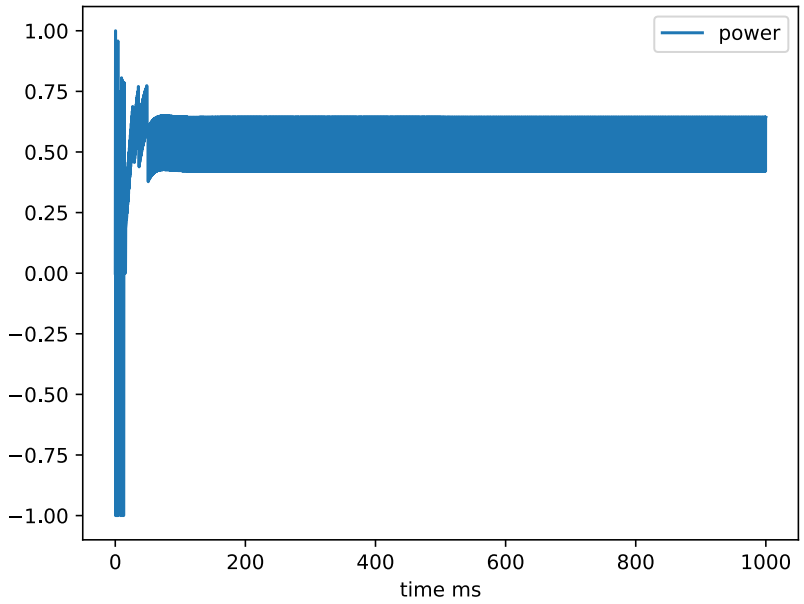

Figure I.176: The amplitude of the normalized output voltage

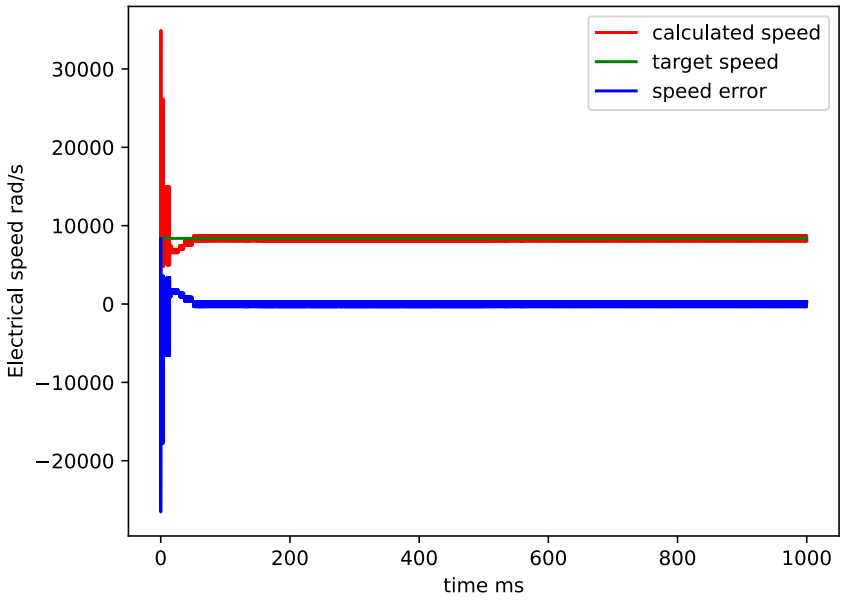

Figure I.177: The calculated, target, and error in the electrical speed of the trapezoidal controller

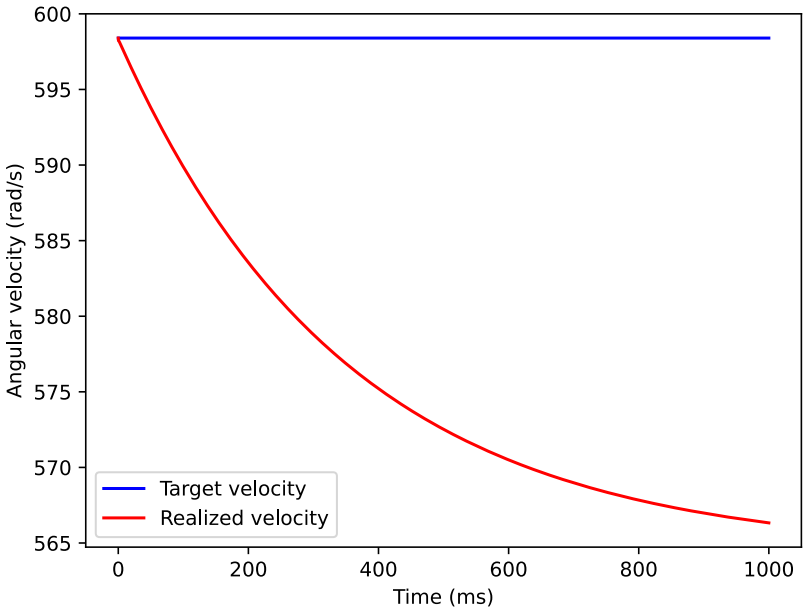

Figure I.178: The desired and realized velocity

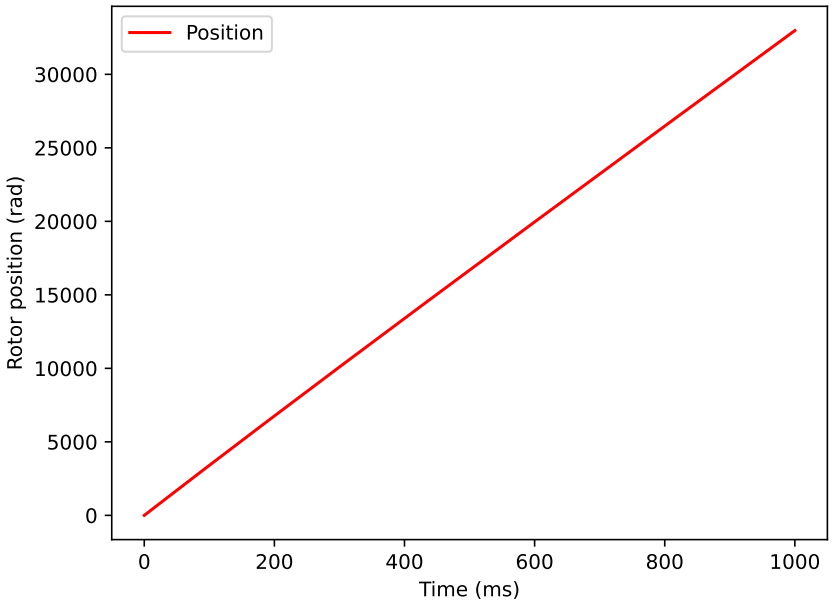

Figure I.179: The position of the rotor

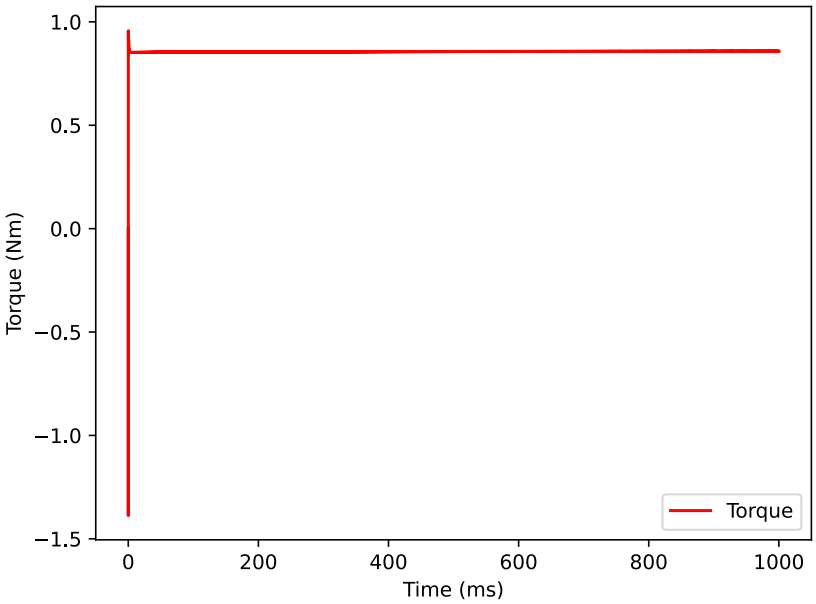

Figure I.180: The torque delivered by the motor

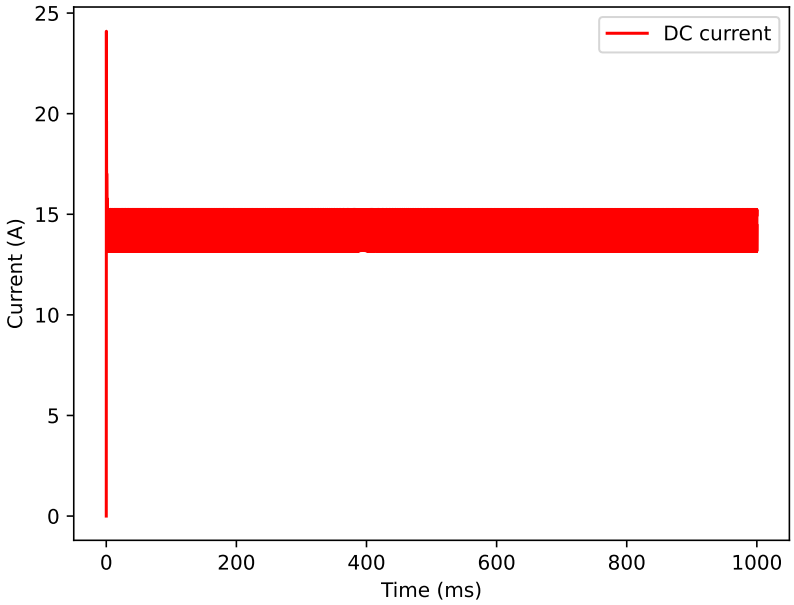

Figure I.181: The DC current through the motor

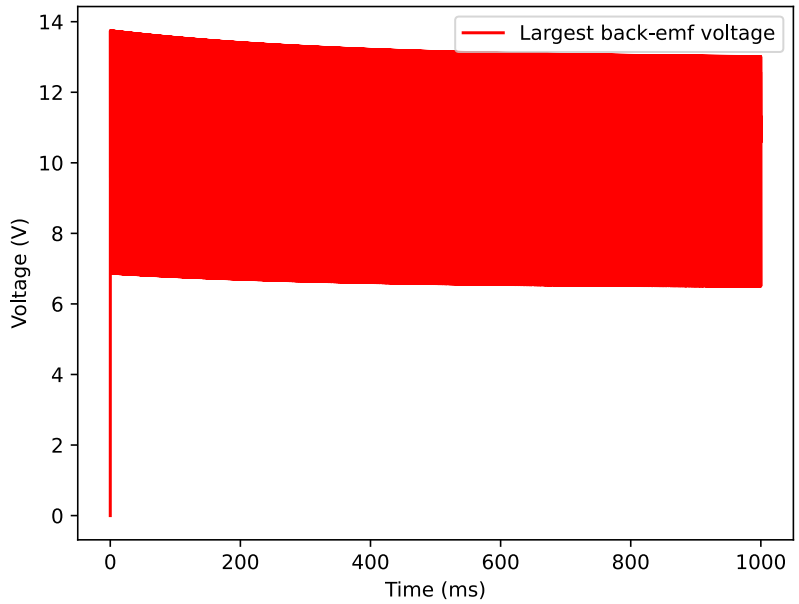

Figure I.182: The largest value for back-emf generated by a motor phase

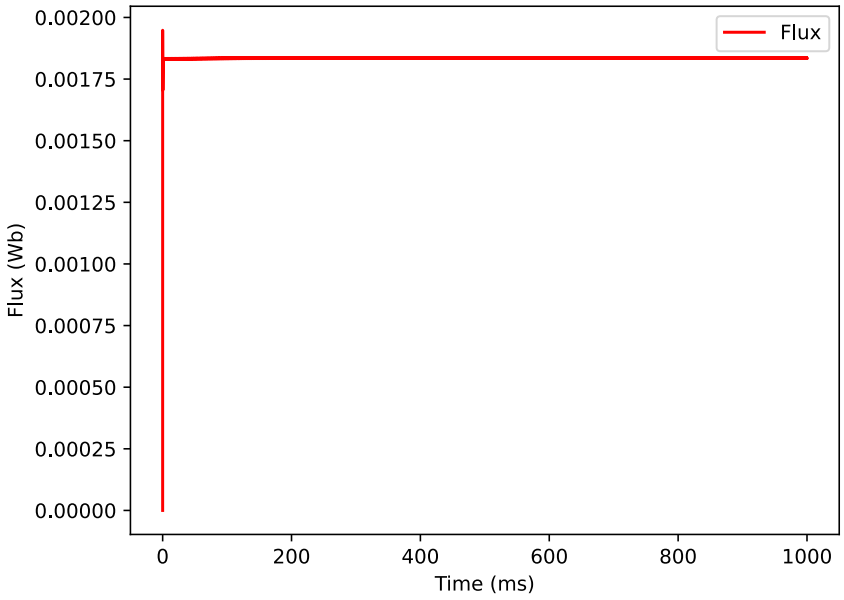

Figure I.183: Flux generated by the motor, for more information see code

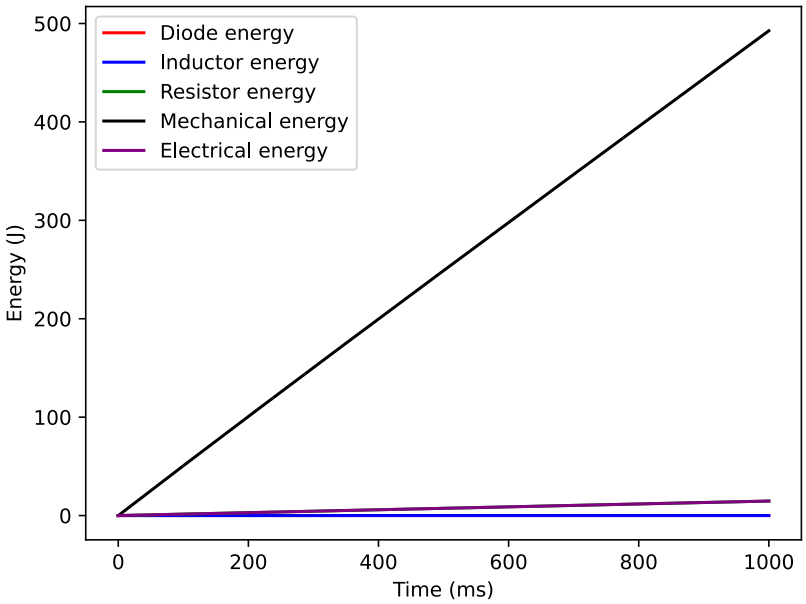

Figure I.184: The energy used or generated by various components, the total electrical energy and the total mechanical energy. The total efficiency for this simulation was 0.971