

Electrical Drives and Electrical Vehicles

Viena Car Simulation

Bernardo Matos 84017
Alejandro Saez 95017
Jamilson Junior 87025
André Gomes 71166

Vienna Project Part4

Table of Contents

Intro	1
dq Model Motor Testing Results	1
FOC Controller	5
Results Track1	7
Results Track2	11
Best values of PI	15
Comparison with Vienna part2	16

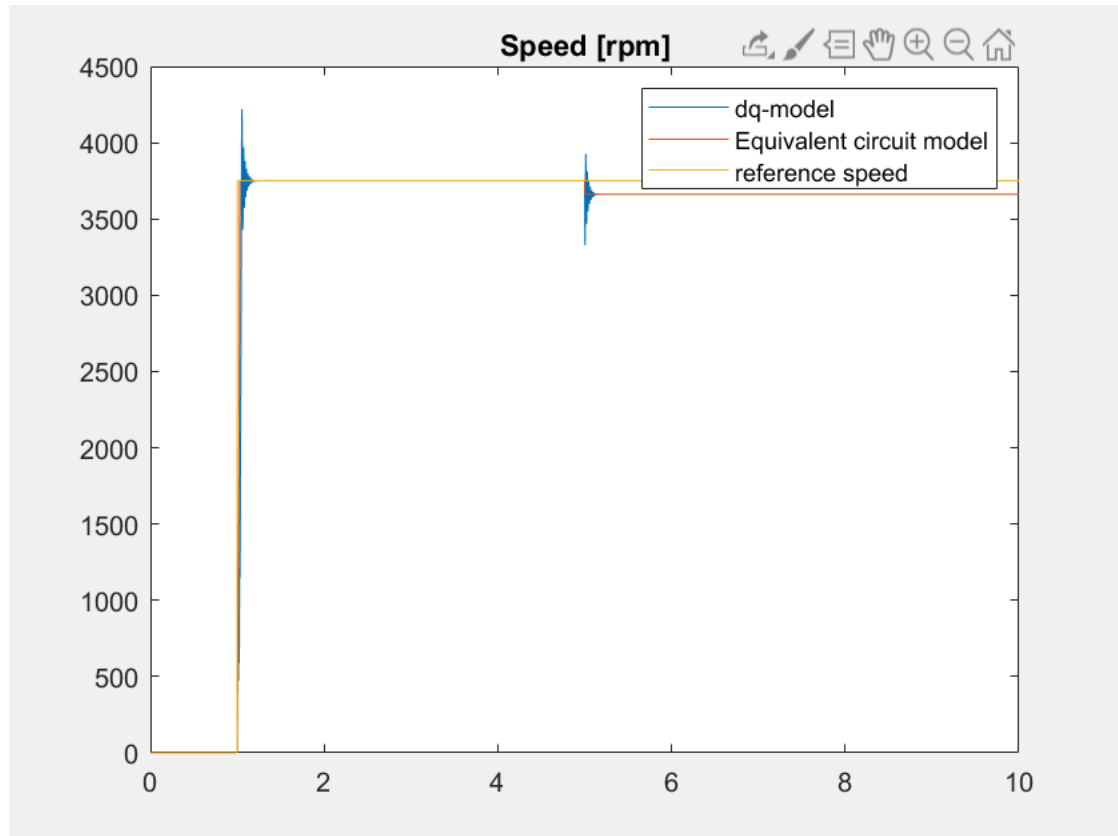
Intro

In this part of the vienna project a new model for the induction motor is studied, the dq model. In order to control this new model, a field oriented control unit was implemented.

```
close all
clear
clc
load(' ./sim_part1.mat');
load('sim_professor_part4');
```

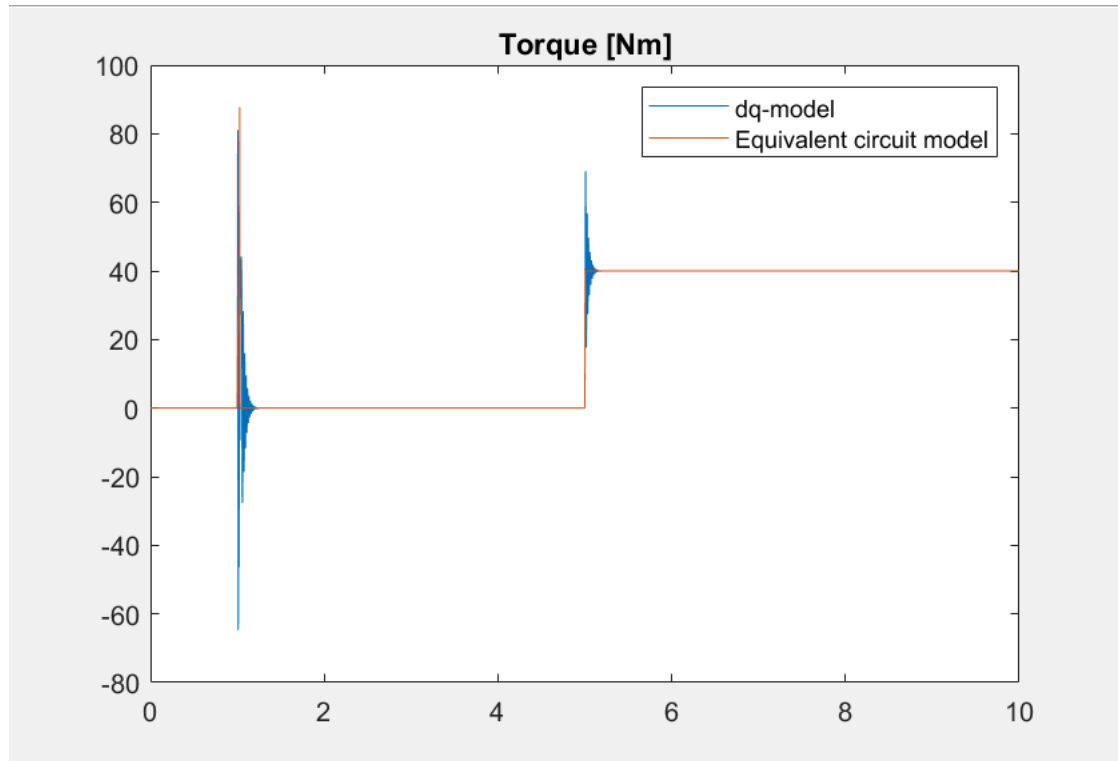
dq Model Motor Testing Results

With the model provided by the professor, the L and the R matrices were filled according with the equations (13) and (14) of the provided laboratory script: For the speed of the motor:



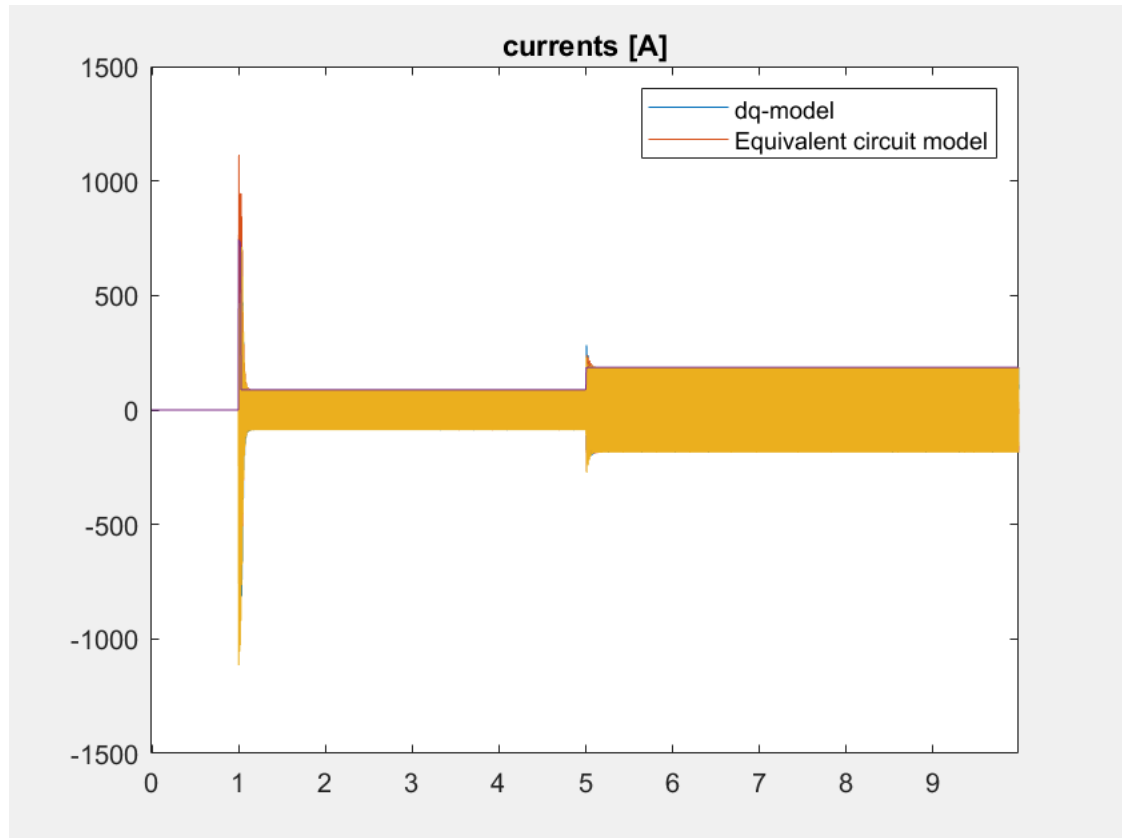
- One can notice that for the steady state conditions, both models follow the reference value of 3750 [rpm], however the dq model shows oscillations in the transient regime, having a maximum of 4220 [rpm] and a minimum on 3430 [rpm], a 12% overshoot.
- When the load is active both models, for steady state conditions stabilize for the same value of 3662 [rpm]. However there is a transient period for the dq model, having a maximum deviation of 9%.

For the torque, the results were:



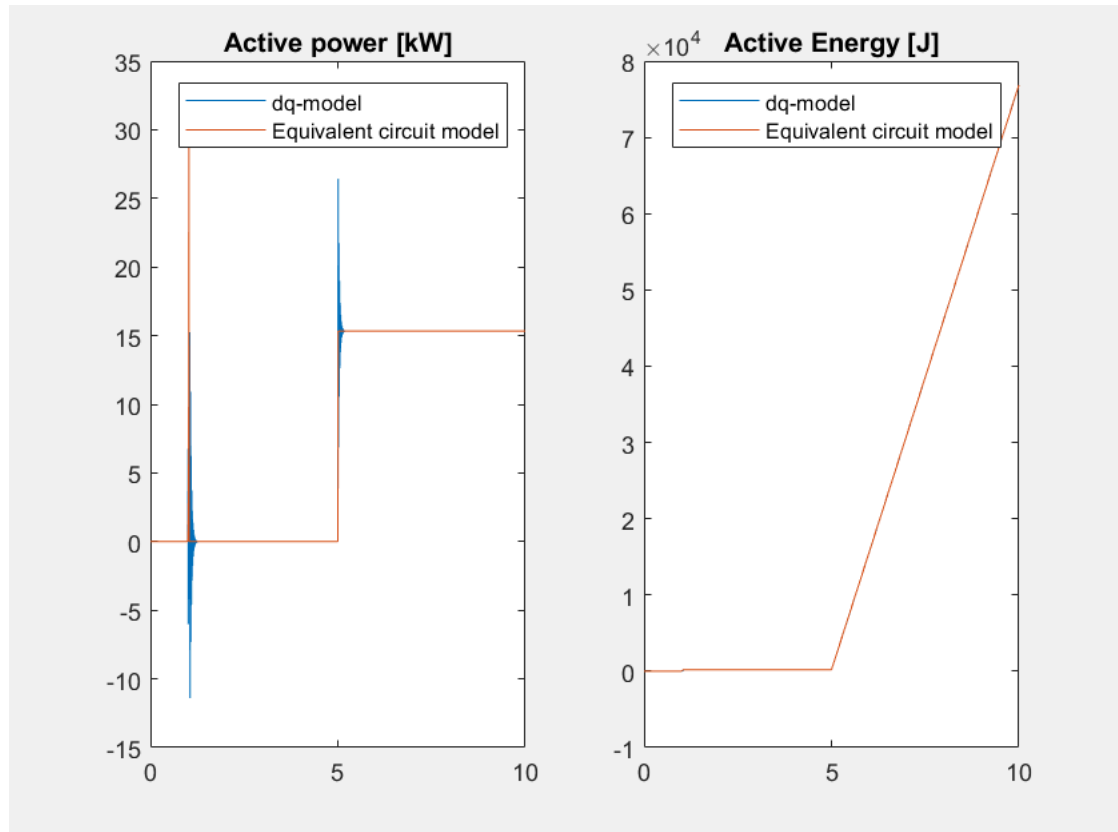
- The starting torque shows a very high peak at the beginning because the machine was stopped and this torque is needed to overcome the inertia of the machine.
- In steady state, both models follow the 40 [Nm] reference.
- In the dq model, there is also a transient, where the maximum value reaches a maximum deviation of 75%.

For the stator current:



- One can see that the start current in the equivalent circuit model only shows one phase, being its maximum value of 741 [A], 8.2 times higher than the steady state current.
- The dq model shows the 3 phases in sinusoidal values, being the maximum starting current of 1114[A], 12.6 times the steady state current.
- In steady state, the equivalent circuit shows only the maximum value of the current, while the dq model shows the 3 phase sinusoidal values.
- With the load, one can see the transient with the 3 phases, having the maximum current 51% higher than the steady state current.

Finally, for the active energy and power:

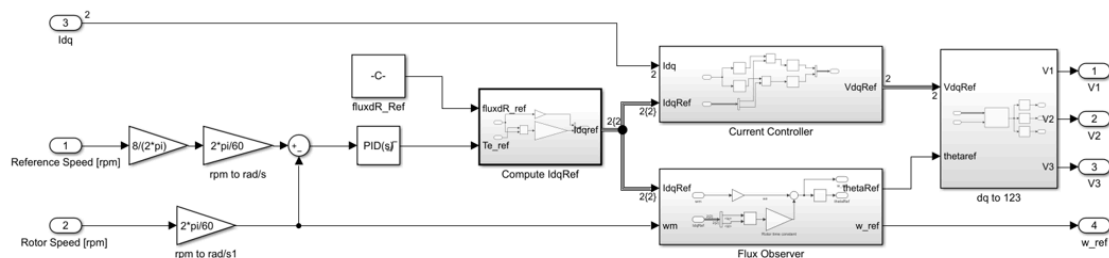


- As expected, the energy in both models are the same.
- For the active power, both models stabilize on the same steady state models, however the dq model shows a transient at the starting and when the load is connected. During starting one can see that there are moments where the induction motor functions as generator, giving energy to the grid, these moments correspond to the same moments where the speed of the motor is greater than the reference speed.

FOC Controller

In order to control the motor, a field oriented control (FOC) was implemented:

```
open_system('vienna_car_part4/FOC/FOC');
```

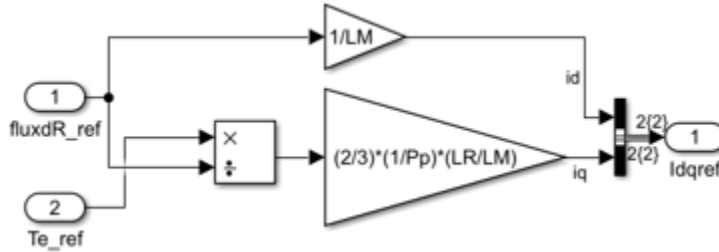


The FOC allows for an independent control of the motor flux and torque, which enables us to use the induction motor in its best conditions.

The FOC functions as follows:

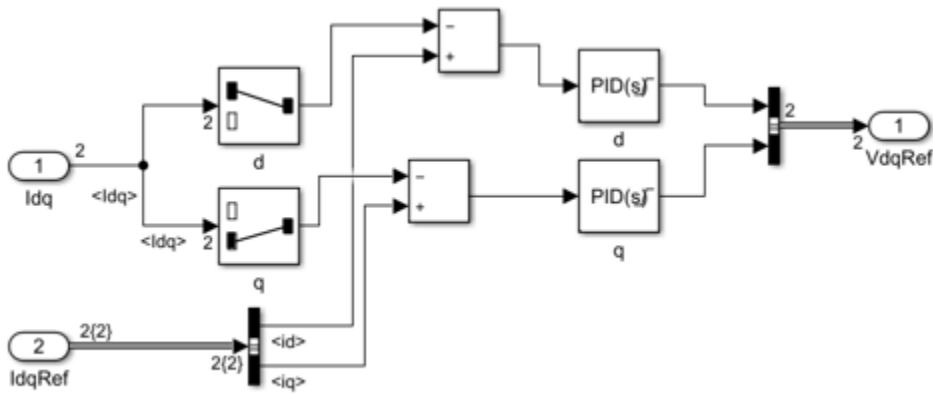
- The input reference and actual motor speed is received, after the difference of these are input to a PI controller where the reference torque is obtained.
- The reference torque, alongside with the rated flux peak to output the Idq reference. This block implements the first equation of the system (21) of the laboratory script resolved with respect to Iq.

```
open_system('vienna_car_part4/FOC/FOC/Compute IdqRef');
```



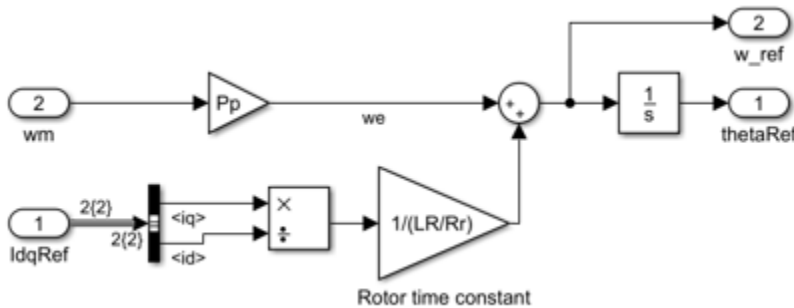
- This Idq reference is input to the current controller, that also receives the actual Idq current to output the stator voltage required to follow the references. This current controller is simply the difference between the reference Idq and the actual Idq applied to a PI controller, that outputs the Vdq needed.

```
open_system('vienna_car_part4/FOC/FOC/Current Controller');
```



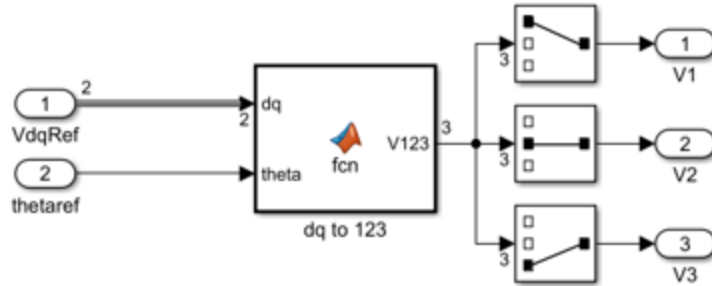
- The Idq reference is also input to the Flux Observer, where alongside with the actual speed of the motor outputs the estimation of the synchronous angle. This block implements the equation (11.85) of the Chapter 11 of Fitzgerald's Electric Machinery.

```
open_system('vienna_car_part4/FOC/FOC/Flux Observer');
```



- Finally, we need to convert the V_{dq} to V_{123} , and for that the estimation of the synchronous angle and the V_{dq} are input to a subsystem where the equation (6) is implemented with respect to V_{123} .

```
open_system('vienna_car_part4/FOC/FOC/dq to 123');
```



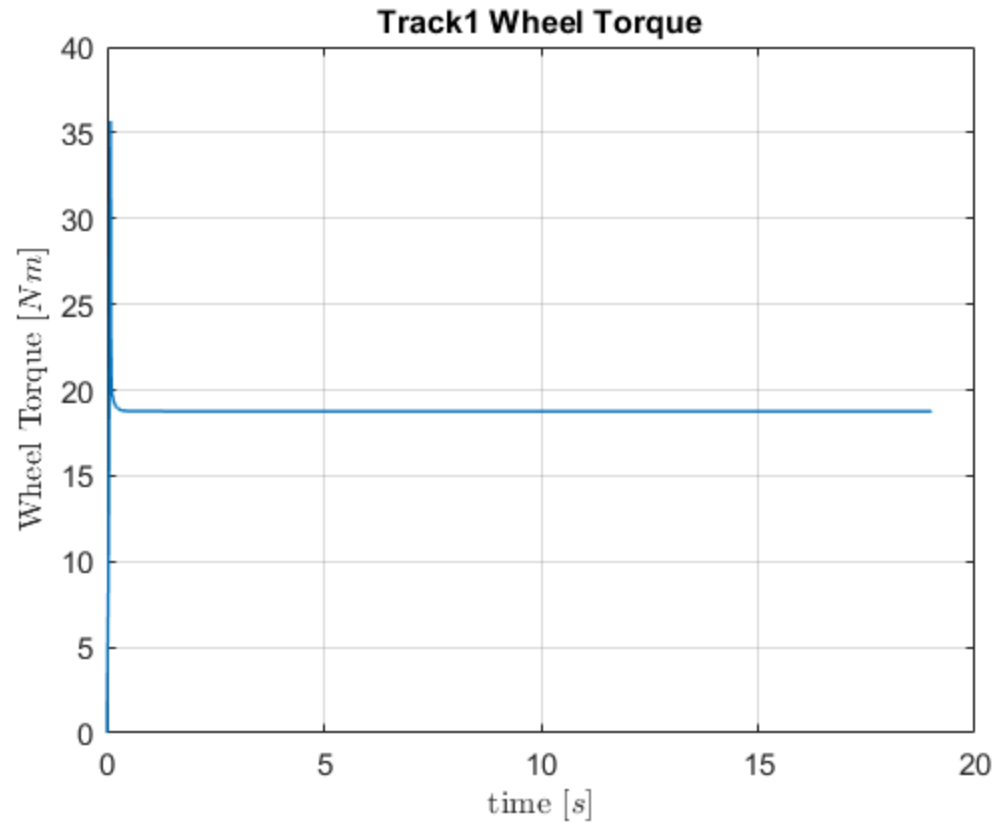
Now, with the FOC complete, one can observe the following results: PI controller has $P = 10$ and $I = 100$.

Results Track1

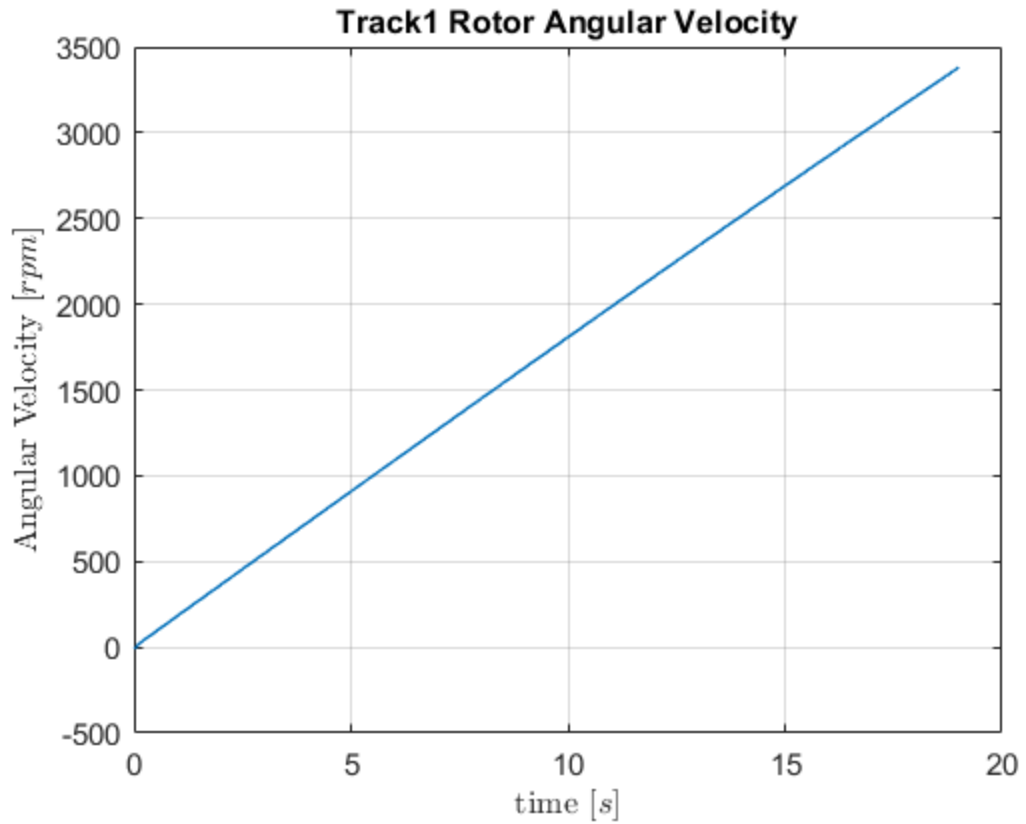
Setting the simulation for track 1 length and slope, the following results are obtained:

```
reference_speed = reference_speed_track1;
slope_of_track = 3;
simulation_length_track = distance_A_B;
TrackAB = sim('vienna_car_part4');

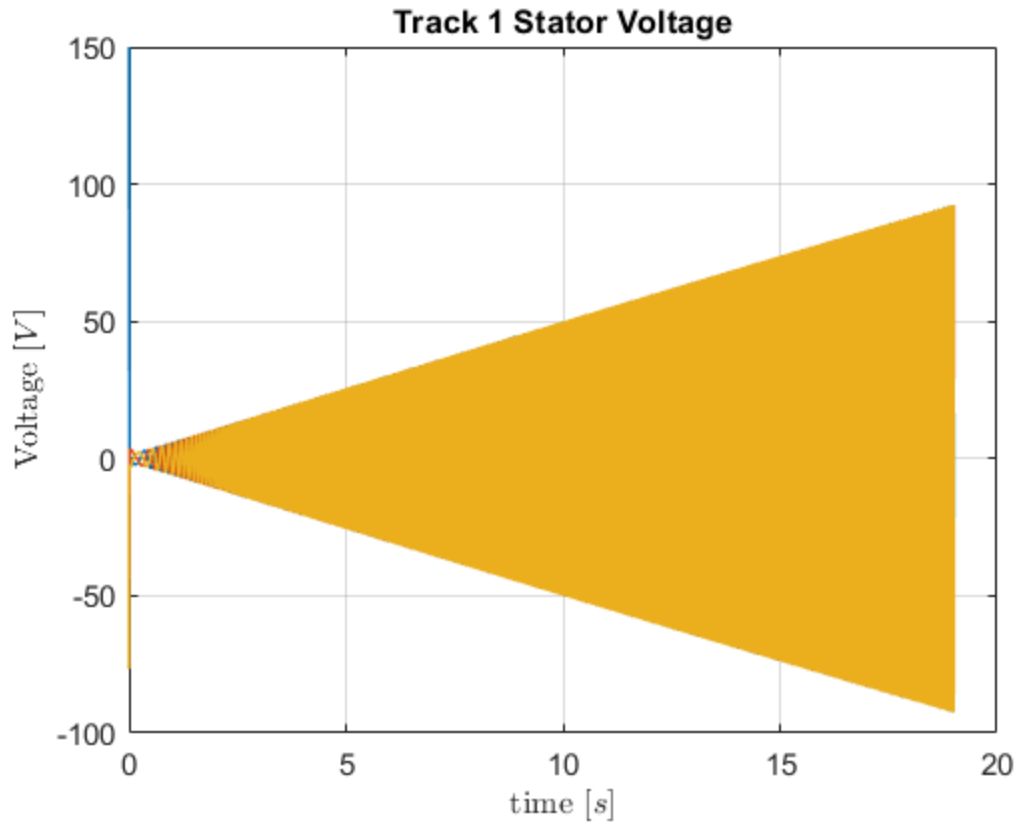
figure()
plot(TrackAB.motor_torque.Time,
     TrackAB.motor_torque.Data, 'LineWidth', 1);
set(gca, 'FontSize', 11);
grid on;
title('Track1 Wheel Torque');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Wheel Torque [Nm]', 'Interpreter', 'latex');
```

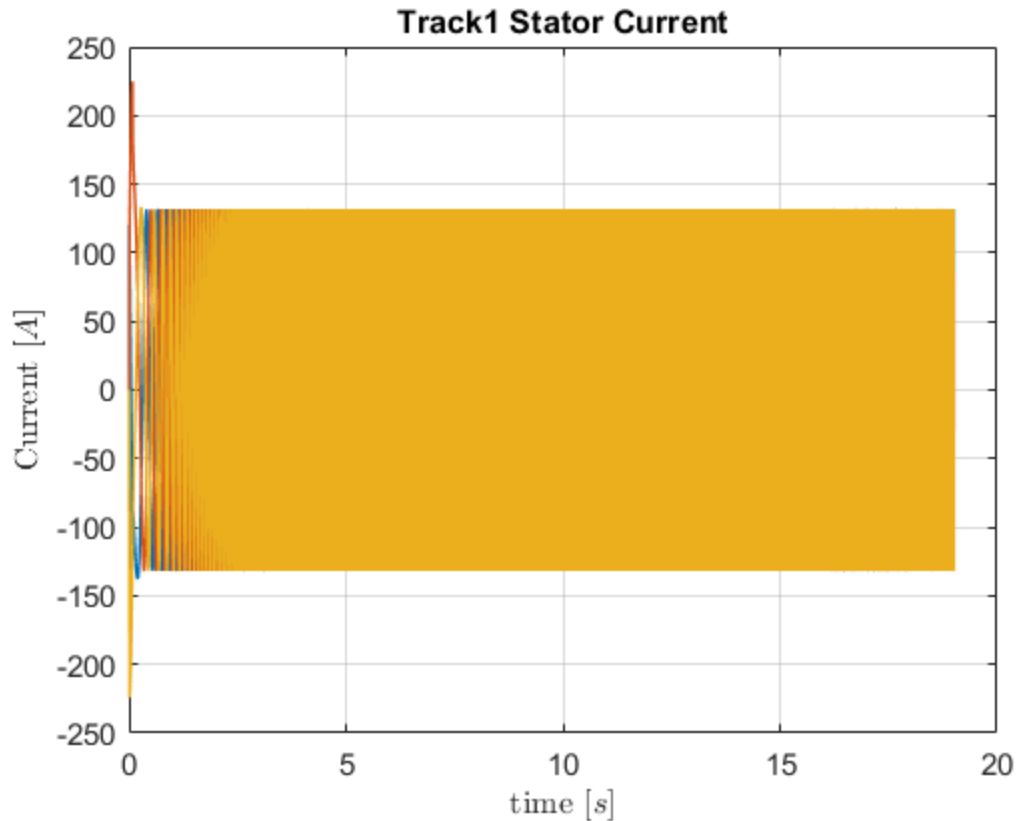
```
figure()
plot(TrackAB.rotor_angular_velocity.Time,
     TrackAB.rotor_angular_velocity.Data, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Track1 Rotor Angular Velocity');
xlabel('time  $[s]$ ', 'Interpreter', 'latex');
ylabel('Angular Velocity  $[rpm]$ ', 'Interpreter', 'latex');
```



```
figure()
plot(TrackAB.stator_voltage.signal1.Time,
     TrackAB.stator_voltage.signal1.Data, 'LineWidth', 1);
hold all
plot(TrackAB.stator_voltage.signal2.Time,
     TrackAB.stator_voltage.signal2.Data, 'LineWidth', 1);
plot(TrackAB.stator_voltage.signal3.Time,
     TrackAB.stator_voltage.signal3.Data, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Track 1 Stator Voltage');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Voltage [V]', 'Interpreter', 'latex');
```



```
figure()
plot(TrackAB.stator_current.Time,
     TrackAB.stator_current.Data, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Track1 Stator Current');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Current [A]', 'Interpreter', 'latex');
```



For the track1, the obtained results show:

- One can see that the torque overshoots for a maximum of 30 [Nm], but later, as expected, stabilizes at 18.75 [Nm] in around 0.6 [sec].
- The motor reaches a maximum of 3381 [rpm].
- One can see that this model allows the visualisation of the three phases of stator voltage. For this track, the maximum voltage increases with time in a constant manner, reaching the maximum of 91.3 [V].
- One can also see the three phases of the stator current. The starting current is quase unstable, and reaches a maximum of 167.5 [A], but later stabilizes in around 132 [A].

Results Track2

Setting the simulation for track 2 length and slope, the following results are obtained:

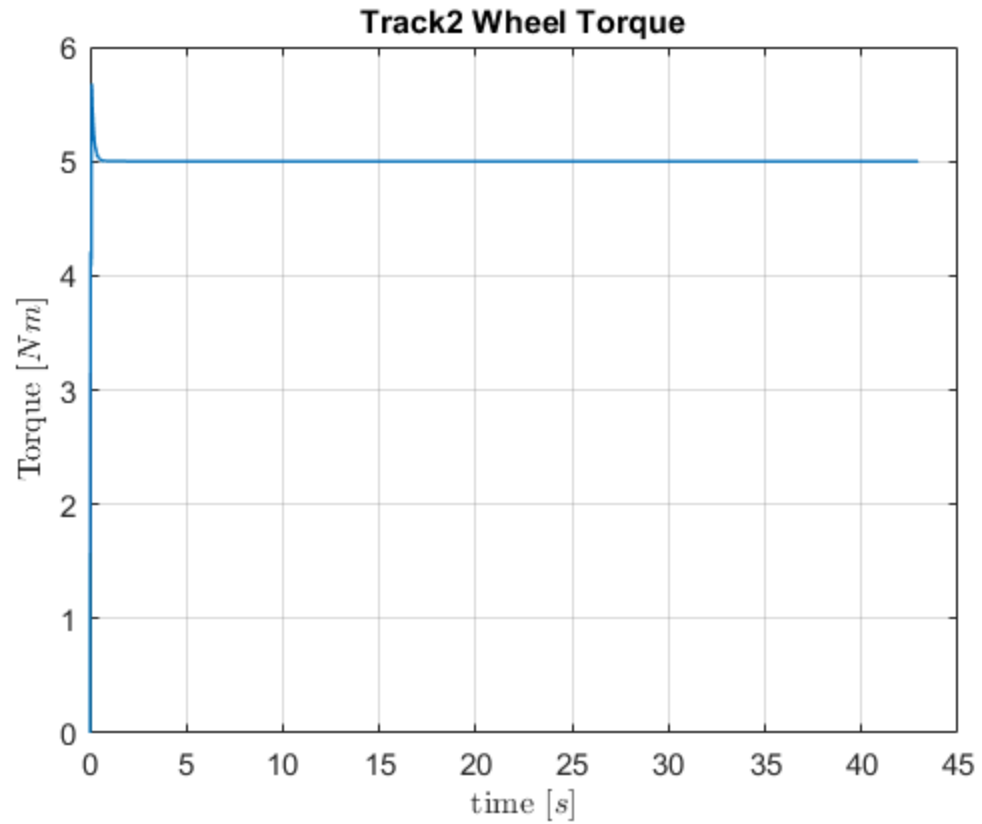
```
reference_speed = reference_speed_track2;
slope_of_track = 0;
simulation_length_track = track2_length;
TrackCDEF = sim('vienna_car_part4');

figure()
plot(TrackCDEF.motor_torque.Time,
     TrackCDEF.motor_torque.Data, 'LineWidth', 1);
set( gca, 'FontSize', 11);
```

```

grid on;
title('Track2 Wheel Torque');
xlabel('time  $[s]$ ','Interpreter','latex');
ylabel('Torque  $[Nm]$ ','Interpreter','latex');

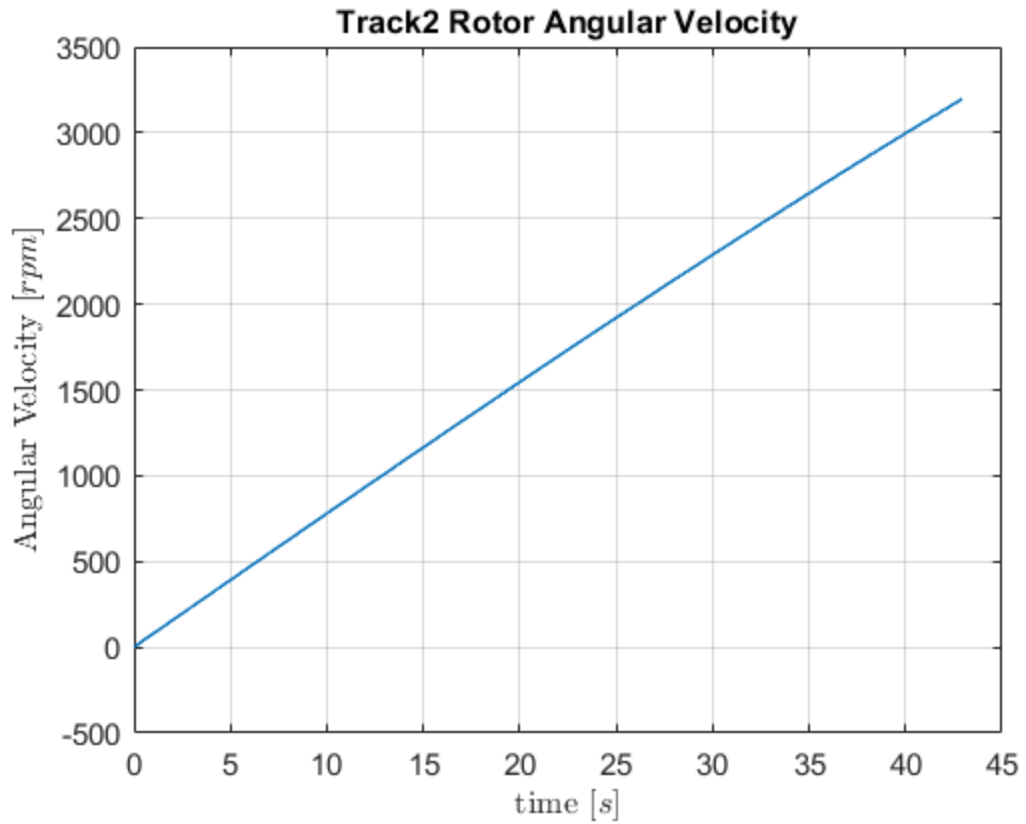
```



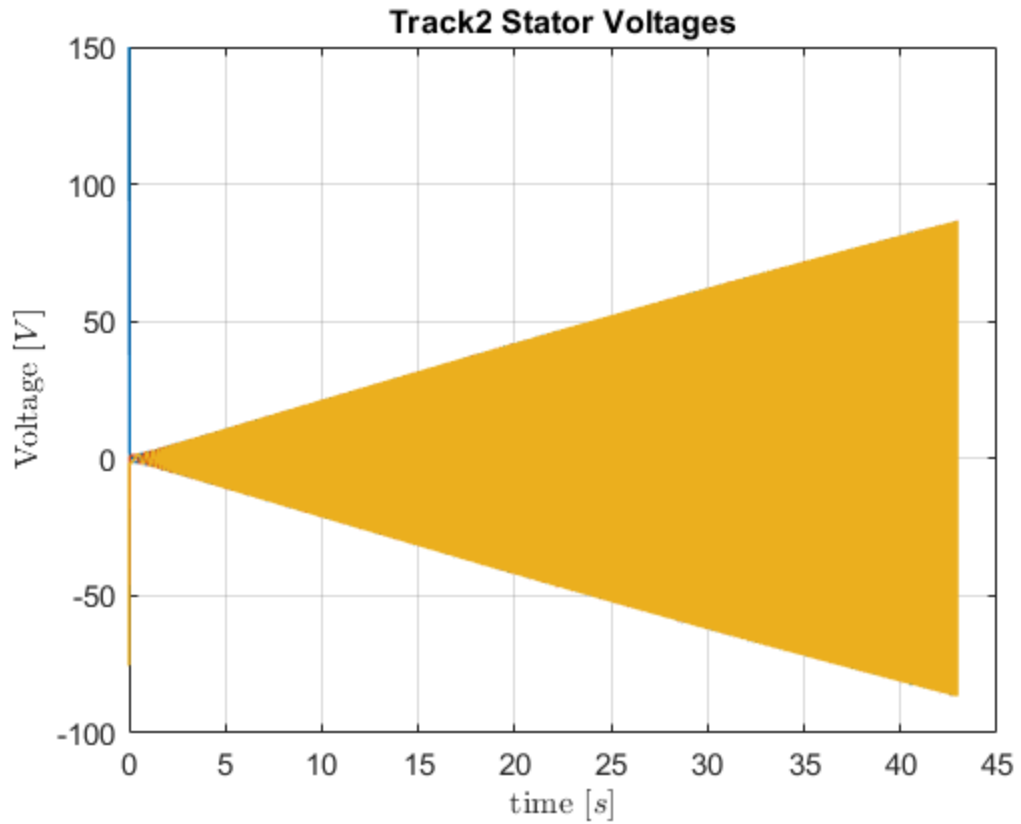
```

figure()
plot(TrackCDEF.rotor_angular_velocity.Time,
     TrackCDEF.rotor_angular_velocity.Data, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Track2 Rotor Angular Velocity');
xlabel('time  $[s]$ ','Interpreter','latex');
ylabel('Angular Velocity  $[rpm]$ ','Interpreter','latex');

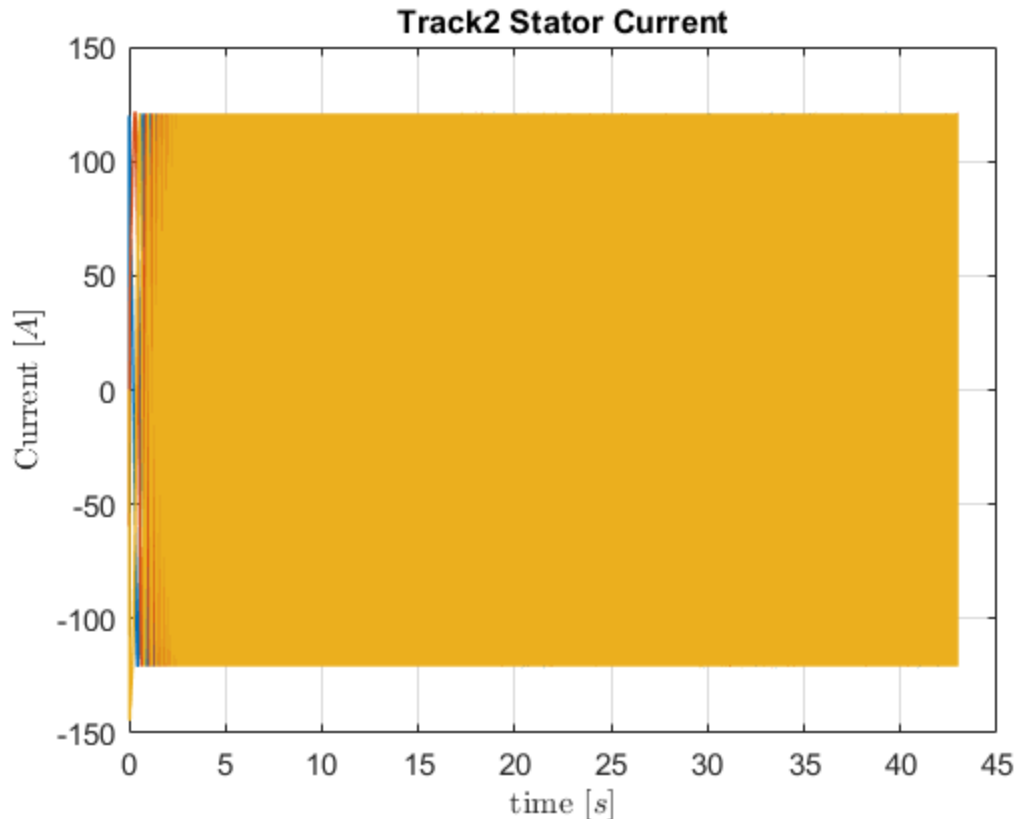
```



```
figure()
plot(TrackCDEF.stator_voltage.signal1.Time,
     TrackCDEF.stator_voltage.signal1.Data, 'LineWidth', 1);
hold all
plot(TrackCDEF.stator_voltage.signal2.Time,
     TrackCDEF.stator_voltage.signal2.Data, 'LineWidth', 1);
plot(TrackCDEF.stator_voltage.signal3.Time,
     TrackCDEF.stator_voltage.signal3.Data, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Track2 Stator Voltages');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Voltage [V]', 'Interpreter', 'latex');
```



```
figure()
plot(TrackCDEF.stator_current.Time,
     TrackCDEF.stator_current.Data, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Track2 Stator Current');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Current [A]', 'Interpreter', 'latex');
```



For the track2, the obtained results show:

- One can see that the torque overshoots for a maximum of 7.2 [Nm], but later, as expected, stabilizes at 5 [Nm] in around 1 [sec].
- The motor reaches a maximum of 3197[rpm].
- One can see that this model allows the visualisation of the three phases of stator voltage. For this track, the maximum voltage increases with time in a constant manner, reaching the maximum of 86.6 [V].
- One can also see the three phases of the stator current. The starting current, for this track, does not overshoot. It stabilizes quickly at around 120[A].

Best values of PI

In order to reach the best values for P and I controllers, the track1 was analysed.

To choose the best value, a try and error method was used, testing P and I with the values of 10, 100, and 200. The following conclusions were achieved:

- For low values of P and increasing, the oscillations, starting torque and starting current.
- Increasing P, makes the system respond faster, eliminates oscillations, but it increases the starting torque and current.
- In conclusion, the best value chosen was $P = 100$ and $I = 100$. The maximum starting current is 224 [A] and the starting torque is 35.6 [Nm], which is less than the maximum motor torque (65 [Nm]).

Comparison with Vienna part2

Comparing with open loop and closed loop controller, respectively, using the track1 torque and stator current:

- The initial torque and current, in open loop controller is of around 25 [Nm] and 165 [A] respectively. It shows an oscillatory behavior, however one must keep in mind that the equivalent circuit is for describing the steady state conditions, what indicates that the real system may be way more oscillatory in the transient state. The FOC, on the other hand, shows no oscillations in the transient, but it shows a higher initial torque and initial current, 35,6 [Nm] and 224 [A] respectively.
- The initial torque and current in the closed loop controller of the Vienna Project part 3, is 22.8 [Nm] and 175 [A] respectively. The initial torque still shows some oscillations during the transient period. The FOC, as said before, has higher starting values, but it is not oscillatory in transient periods.

The most important difference between these two controllers is that, although a VFD is relatively simpler to implement than the FOC, by controlling the torque with the values of stator voltage and frequency one also changes the flux of the machine. The FOC in the other hand allows for a independent controlling of the machine's flux and torque, thus one can set the flux for is rated peak value while varying the torque. The dq model used in this laboratory also allows for a better analysis of the phase currents and voltages as well as the transient periods.

Published with MATLAB® R2019b