

Electrical Drives and Electrical Vehicles

Viena Car Simulation

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

Vienna Car part2

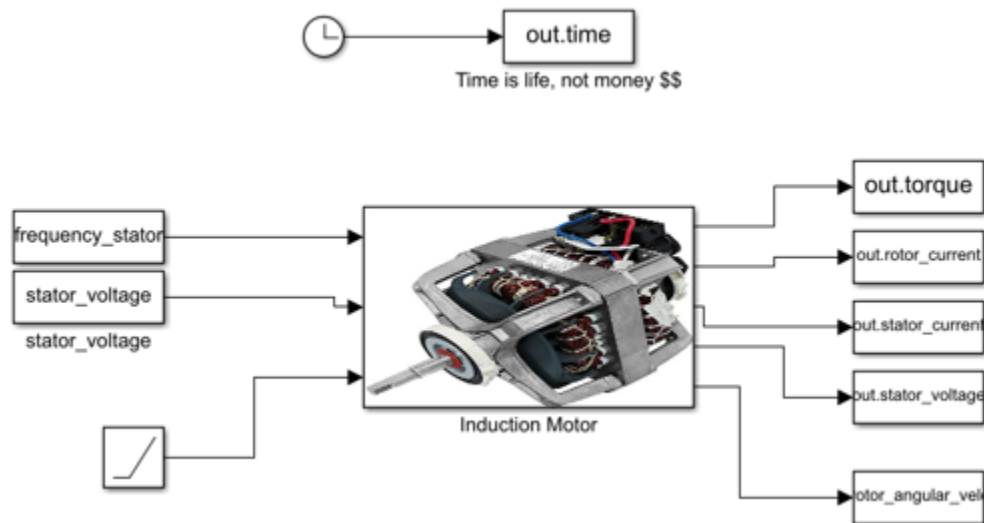
Table of Contents

Induction Motor	1
Induction Motor Model Characteristics	1
Initial and Maximun Torque	5
Car Dynamics and Induction Motor joint System	8
Joint System: Track1	9
Track 1 and Track 2 Comparisons	21
Conclusions	21

Induction Motor

In this part of the Vienna Car project, the behavior of the induction motor will be simulated.

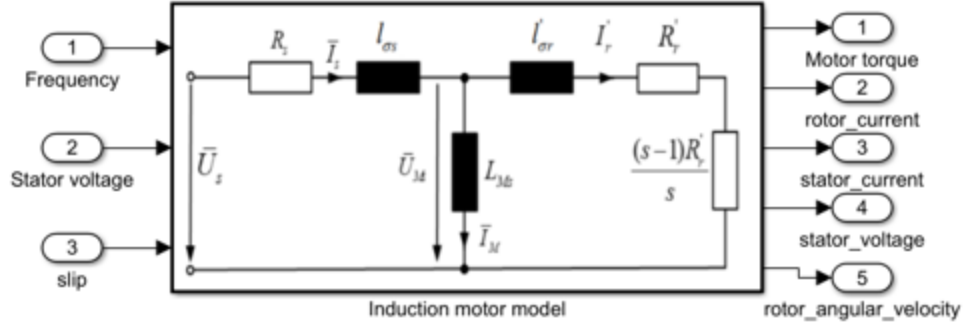
```
open_system('induction_motor');
```



Induction Motor Model Characteristics

From the motor characteristic constants given in the project script, one can calculate the impedances, currents and voltages of the circuit.

```
open_system('induction_motor/Induction Motor');
```



From the KCL one can obtain that:

$$I_s = I_M + I_R$$

The stator current is given by the stator voltage divided by the equivalent impedance of the circuit:

$$I_s = \frac{V_s}{(R_s + j\omega l_{\sigma s}) + (j\omega L_{Ms}) // (\frac{(s-1)R_r}{s} + R_r + j\omega l_{\sigma r})}$$

The rotor current is given by:

$$I_r = I_s - \frac{V_M}{j\omega L_{Ms}}$$

Applying the KVL to the left circuit loop, one can obtain V_{Ms} as:

$$V_{Ms} = V_s - I_s Z_s$$

Knowing now the rotor current, now it is possible to find the mechanical power output of the machine:

$$P_{mec} = 3I_r^2 \frac{R_r(s-1)}{s}$$

Notice that the power output of the machine is three times the power output of each phase.

Finally, the torque is given by the ratio between the mechanical power and the rotor's angular velocity:

$$T = \frac{P_{mec}}{s\Omega_s}$$

The isolated induction motor was implemented in the following manner:

```
function [torque, rotor_current, stator_current, stator_voltage,
rotor_angular_velocity] = fcn(stator_frequency, stator_voltage, slip)
% Constant specs
stator_resistance = 8.56*1e-3;
stator_leakage_inductance = 0.06292 * 1e-3;
mutual_inductance = 1.0122 * 1e-3;
rotor_resistance = 10.20 *1e-3;
rotor_leakage_inductance = 0.06709 * 1e-3;
% Variable specs
```

```
    stator_voltage = stator_voltage/sqrt(3); % phase-phase -> % phase-
neutral
    stator_reactance = stator_leakage_inductance * 2 * pi *
stator_frequency * li;
    rotor_reactance = rotor_leakage_inductance * 2 * pi *
stator_frequency * li;
    mutual_reactance = mutual_inductance * 2 * pi * stator_frequency *
li;
    stator_impedance = stator_resistance + stator_reactance;
    rotor_impedance = rotor_resistance + rotor_reactance;
    % Descriptive Equations
    equivalent_impedance = stator_impedance + (mutual_reactance*(((slip
+1)/slip)*rotor_resistance+rotor_impedance))/(mutual_reactance+(((slip+1)/slip)*rotor_resistance+rotor_impedance));
    stator_current = stator_voltage/equivalent_impedance;
    % Calculus of torque
    rotor_current = stator_current - (stator_voltage-
stator_current*stator_impedance)/(mutual_reactance);
    if (slip == 0 )
        torque = 0;
    else
        torque = (3*rotor_resistance*abs(rotor_current)^2)/
(slip*stator_frequency*2*pi); %if s = 0 => binario igual a 0
    end
    rotor_angular_velocity = (1-slip)*stator_frequency*60;
end
```

So, from the simulink model of the induction motor, one can see the lines of torque for different stator voltage and frequencies, being $V/f = 1$:

```
clear
clc
close all
```

In order to understand the behavior of the torque for several $V/f = 1$ values, a simulation of the induction motor will be made with the following inputs.

```
frequency_stator_vector = 10:1:76;
stator_voltage_vector = 10:1:76;
```

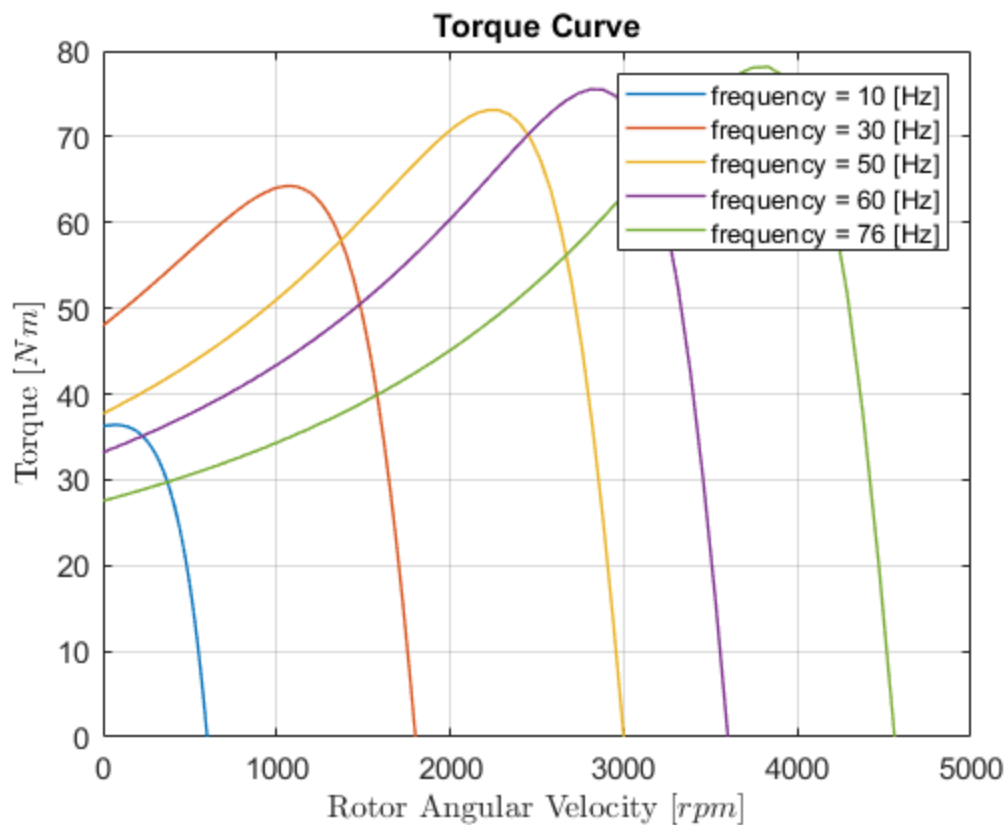
The resulted response is as follows:

```
figure();
% Maximum torques
max_torque_values = [];
% Initial torque
init_torque_values = [];
for i = 1:length(frequency_stator_vector)
    frequency_stator = frequency_stator_vector(i);
    stator_voltage = stator_voltage_vector(i);
    induction_motor_simulation = sim('induction_motor');
    max_torque_values = [max_torque_values
max(induction_motor_simulation.torque)]; % Save for later
    init_torque_values = [init_torque_values
induction_motor_simulation.torque(1)]; % Save for later
```

```

    if (frequency_stator_vector(i) == 10 || frequency_stator_vector(i)
    == 30 || frequency_stator_vector(i) == 50 ||
    frequency_stator_vector(i) == 60 || frequency_stator_vector(i) ==
    76 )
        plot(induction_motor_simulation.rotor_angular_velocity ,
induction_motor_simulation.torque, 'LineWidth', 1);
        hold all
        set( gca, 'FontSize', 11);
        grid on;
        title('Torque Curve');
        xlabel('Rotor Angular Velocity
$[rpm]$', 'Interpreter', 'latex');
        ylabel('Torque [$Nm]$', 'Interpreter', 'latex');
    end
end
legend('frequency = 10 [Hz]', 'frequency = 30 [Hz]', 'frequency = 50
[Hz]', 'frequency = 60 [Hz]', 'frequency = 76 [Hz]');

```



VFD are able to vary the operating speed of the motor by changing the electrical frequency input to the motor. The speed an AC induction motor operates is given by the following equation: Synchronous Speed = $120 \times \text{Frequency} / \text{Number of Poles}$.

This shows that for lower values of frequency, the Torque lowers as well, this is because for lower values of frequency the Stator impedance starts to influence more the flux, because it's voltage drop increases in relation to the stator voltage. This is not ideal and another not ideal thing to notice is that the curve translations to the left as the frequency decreases are not shifting with constant spacing.

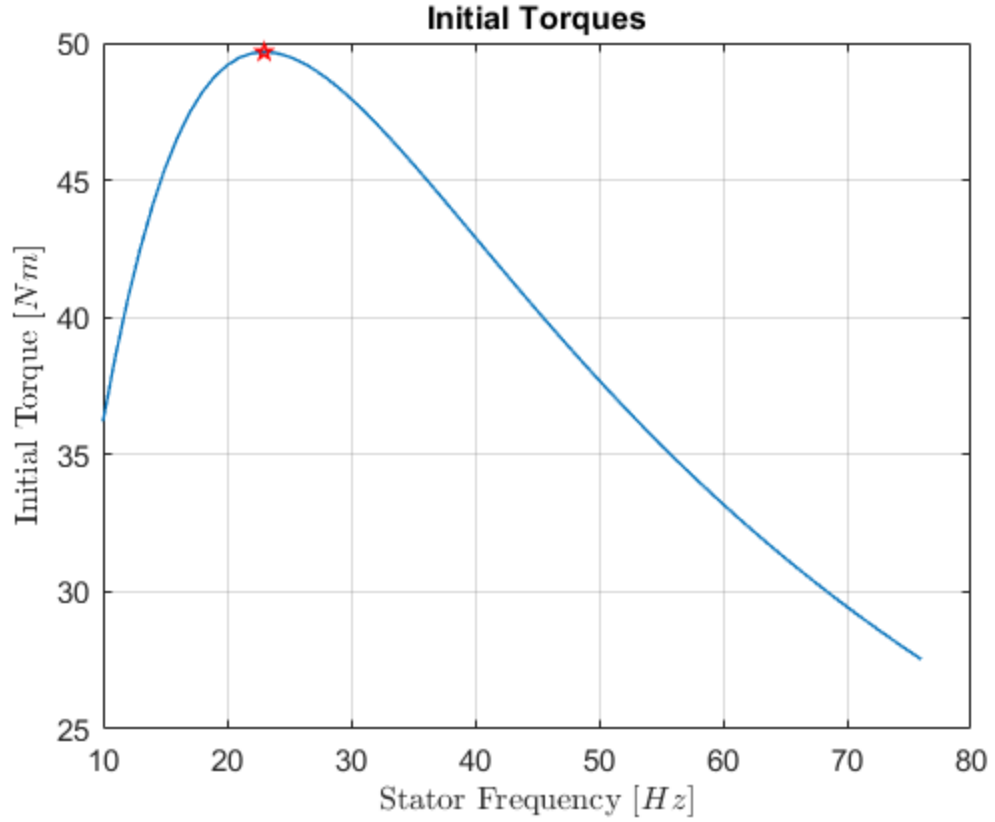
VFD actually controls both frequency and voltage simultaneously to maintain a constant volts/hertz ratio which keeps current flow similar to full speed conditions. This allows the motor to draw full current at any speed and produce full torque as motor speed changes. Increasing frequency above 60 hertz makes the motor run faster than normal and creates two primary concerns: - Many motors and devices are not mechanically balanced to operate at increased speeds and will create vibration, mechanical and safety problems. - In order to maintain a constant horsepower output to drive our load, if speed is increased, torque must decrease! At some point of increased speed we may not be able to produce enough torque to drive the load and at this point, the motor will slow even with increasing frequency. This point is different for each manufacturer's motor and dependent on the torque required by the load.

Initial and Maximun Torque

Now, it is interesting to study what are the $V/f = 1$ values that possess the highest initial and maximum torque, in order to discover what are the best operation conditions for the induction motor. To do that, one must only sweep through the various values of $V/f = 1$, and then plot the values initial and maximum torque.

With the values calculated computed before, one can plot the follow behavior:

```
[max_init_torque, frequency] = max(init_torque_values);
figure()
plot(frequency_stator_vector, init_torque_values,
frequency_stator_vector(frequency), init_torque_values(frequency), 'pr', 'LineWidth
1);
set( gca, 'FontSize', 11);
grid on;
title('Initial Torques');
xlabel('Stator Frequency $[Hz]$', 'Interpreter', 'latex');
ylabel('Initial Torque [$Nm$]', 'Interpreter', 'latex');
max_initual_torque_value = sprintf('The maximum initial torque
is given by: frequency = %d [Hz] , Initial Torque = %d [Nm]',
frequency_stator_vector(frequency), init_torque_values(frequency));
```



Obtaining the following result:

```
fprintf(max_initial_torque_value);
```

The maximum initial torque is given by: frequency = 23 [Hz] , Initial Torque = 4.967301e+01 [Nm]

Looking at the expression used for calculation the torque:

$$T = \frac{3r_r I_{eff}^2}{s\Omega_s}$$

One can predict the behavior of the maximum initial torque by analyzing the first and second derivatives of the torque with respect to the frequency:

$$\frac{d}{df} \frac{3r_r I_{eff}^2}{s\Omega_s} = \frac{3r_r}{2\pi f^2}$$

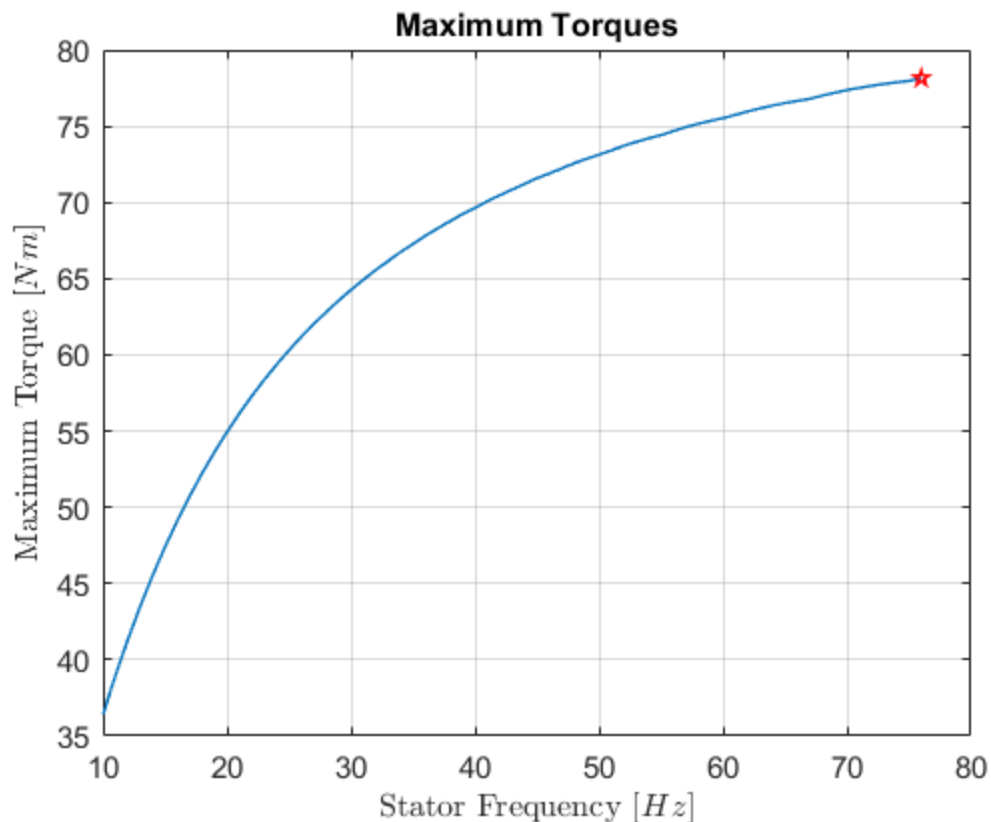
The first delivery indicates that, with $s = 1$, and for positive frequencies, the maximum torque will always be positive.

$$\frac{d}{df} \frac{3r_r}{2\pi f^2} = -\frac{3r_r}{\pi f^3}$$

What indicates that, for positive values of frequency, the maximum torque behaviour will be a parabola with concavity facing down.

The same approach can be taken for the Maximum torque:

```
[max_torque, frequency] = max(max_torque_values);
figure()
plot(frequency_stator_vector, max_torque_values,
frequency_stator_vector(frequency),max_torque_values(frequency),'pr', 'LineWidth
1);
set( gca, 'FontSize', 11);
grid on;
title('Maximum Torques');
xlabel('Stator Frequency [Hz]','Interpreter', 'latex');
ylabel('Maximum Torque [Nm]','Interpreter', 'latex');
max_torque_value = sprintf('The maximum torque is given
by: frequency = %d [Hz] , Initial Torque = %d [Nm]',
frequency_stator_vector(frequency), max_torque_values(frequency));
```



Obtaining the following result:

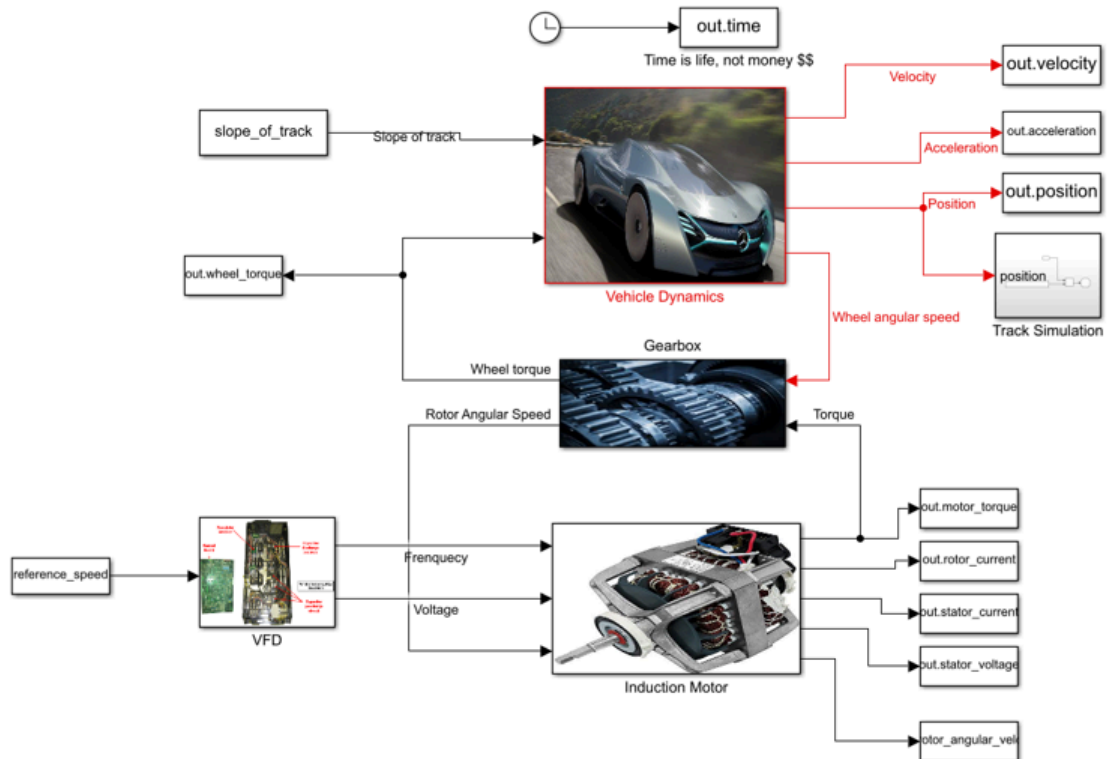
```
fprintf(max_torque_value);
```

The maximum torque is given by: frequency = 76 [Hz] , Initial Torque = 7.815393e+01 [Nm]

As expected, one can see that the maximum torque increases with the frequency. Given that $V/f=1$, a greater frequency implies a greater stator voltage, which due to the torque expression used above, implies a greater rotor current, thus resulting in a greater torque.

Car Dynamics and Induction Motor joint System

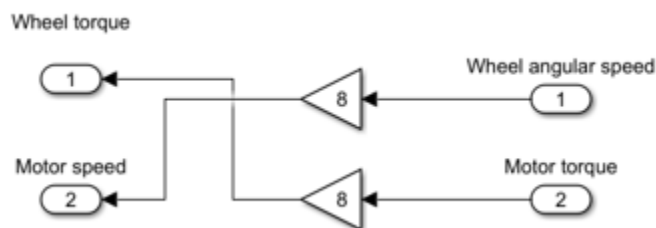
```
open_system('vienna_car_part2');
```



After the study of the induction motor working isolated, it is interesting to connect the first and the second part of the Vienna Car Project.

The induction motor will provide the vehicle dynamics system with torque, and the vehicle dynamics will provide the induction motor with the wheel speed. In order to have the induction motor working at higher speeds (for better efficiency), a gearbox with a ratio of 1:8 is used.

```
open_system('vienna_car_part2/Gearbox');
```



As one can see in the image above:

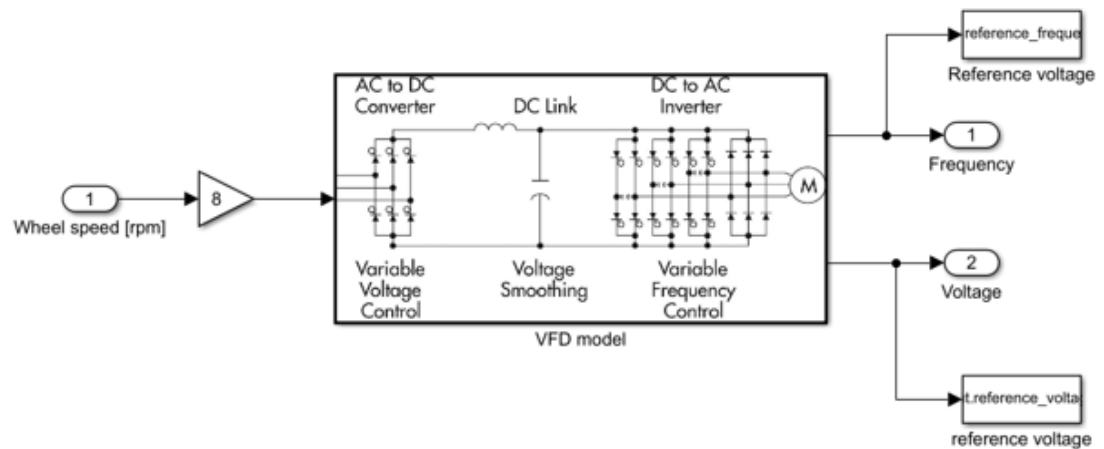
- The gearbox multiplies the motor torque by 8, giving the wheels torque;

- The gearbox multiplies the wheel angular speed by 8, giving the rotor speed;

With the joint system, it is now necessary to provide the induction motor with a reference speed. The reference speed will be given by the speed of the wheels developed in the first part of the Vienna Project.

The wheels angular speed will be transformed in the reference speed by the VFD model created bellow:

```
open_system('vienna_car_part2/VFD');
```



```
function [frequency, voltage] = fcn(reference_speed)
    frequency = (reference_speed*2)/60;
    voltage = frequency;
end

clc
load(' ../vienna_car_part1/sim_part1');
close all
```

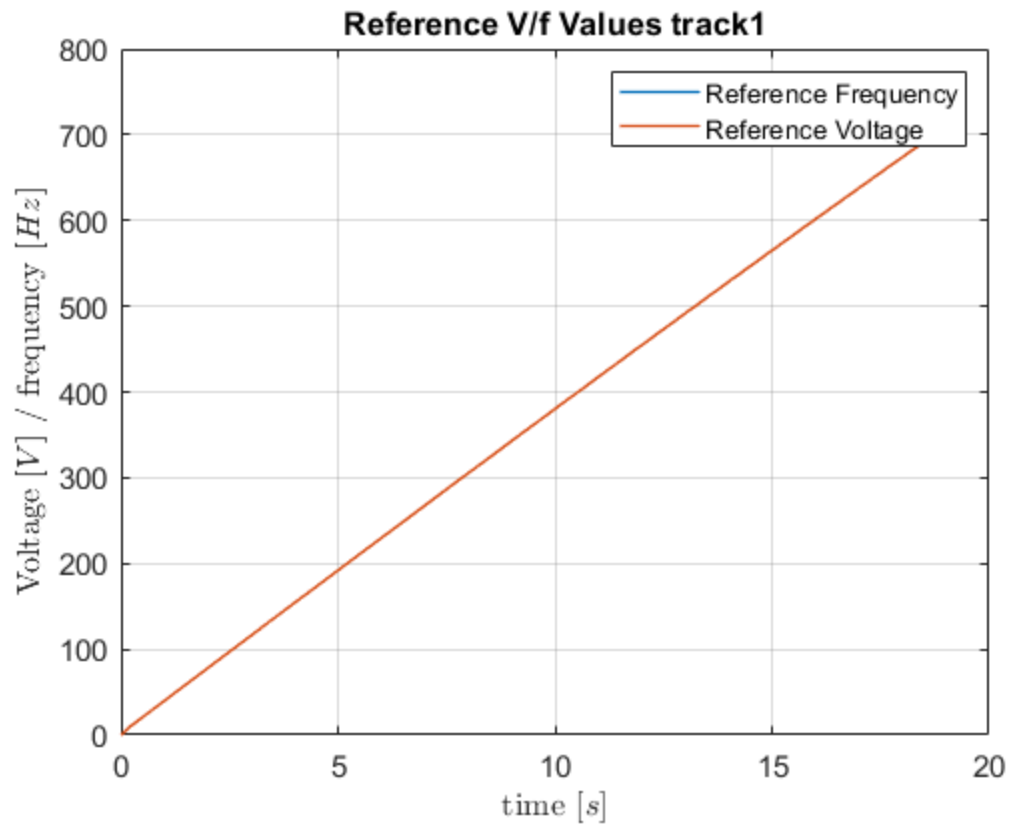
Joint System: Track1

```
slope_of_track = 3;
reference_speed = reference_speed_track1;
simulation_length_track = distance_A_B;
track1_part2 = sim('vienna_car_part2');
```

The reference V/f value for this track is given by:

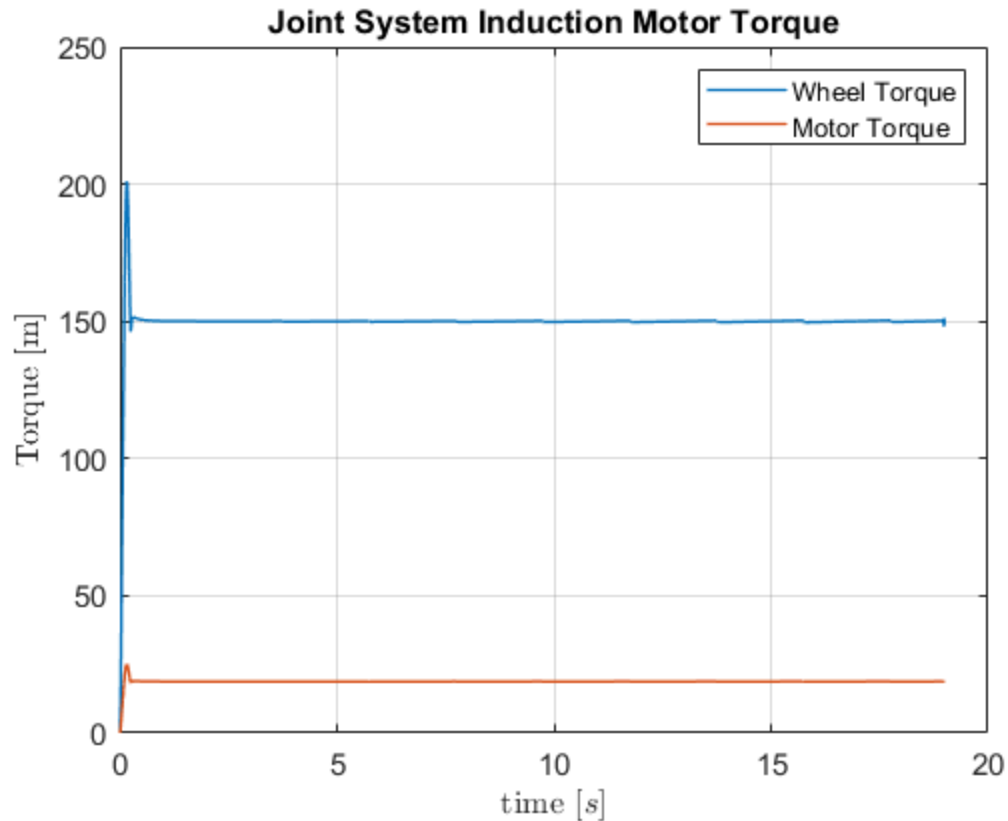
```
figure()
plot(track1_part2.time, track1_part2.reference_frequency, 'LineWidth',
    1);
hold all
plot(track1_part2.time, track1_part2.reference_voltage, 'LineWidth',
    1);
set(gca, 'FontSize', 11);
grid on;
title('Reference V/f Values track1');
xlabel('time [s]', 'Interpreter', 'latex');
```

```
ylabel('Voltage  $[V]$  / frequency  $[Hz]$ ','Interpreter','latex');
legend('Reference Frequency', 'Reference Voltage')
```



The resultant wheel torque is given by:

```
figure()
plot(track1_part2.time, track1_part2.wheel_torque, 'LineWidth', 1);
hold all;
plot(track1_part2.time, track1_part2.motor_torque, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Joint System Induction Motor Torque');
xlabel('time  $[s]$ ','Interpreter','latex');
ylabel('Torque  $[m]$ ','Interpreter','latex');
legend('Wheel Torque', 'Motor Torque');
```



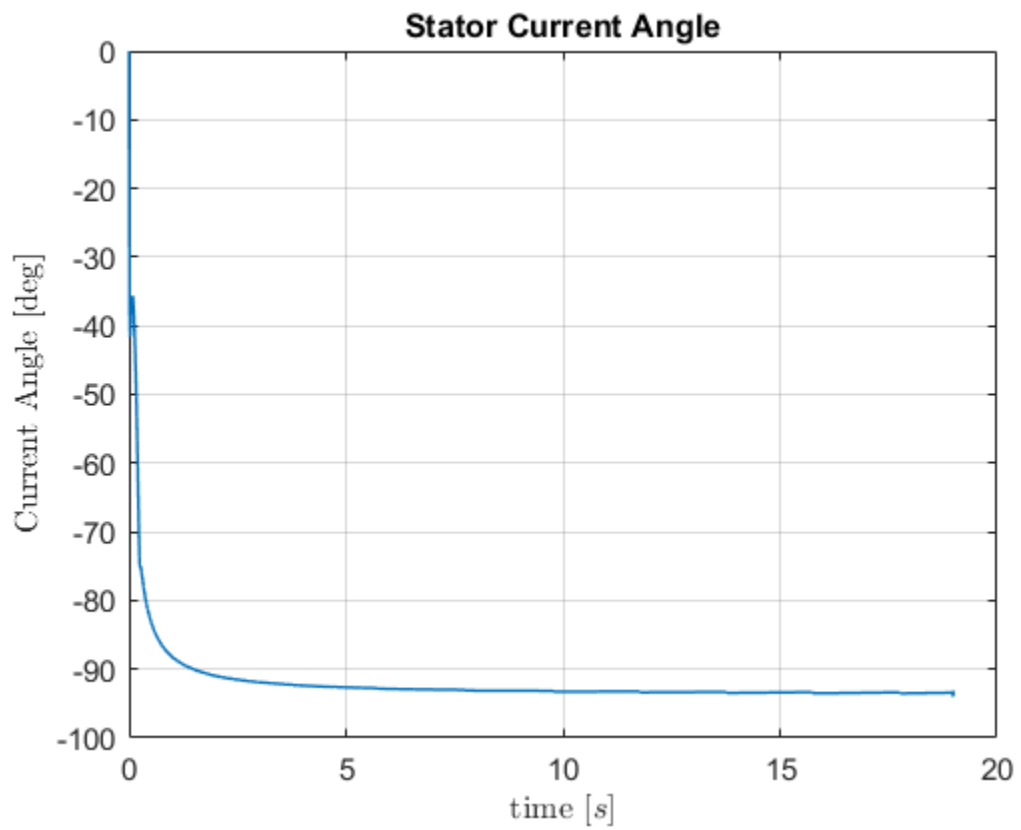
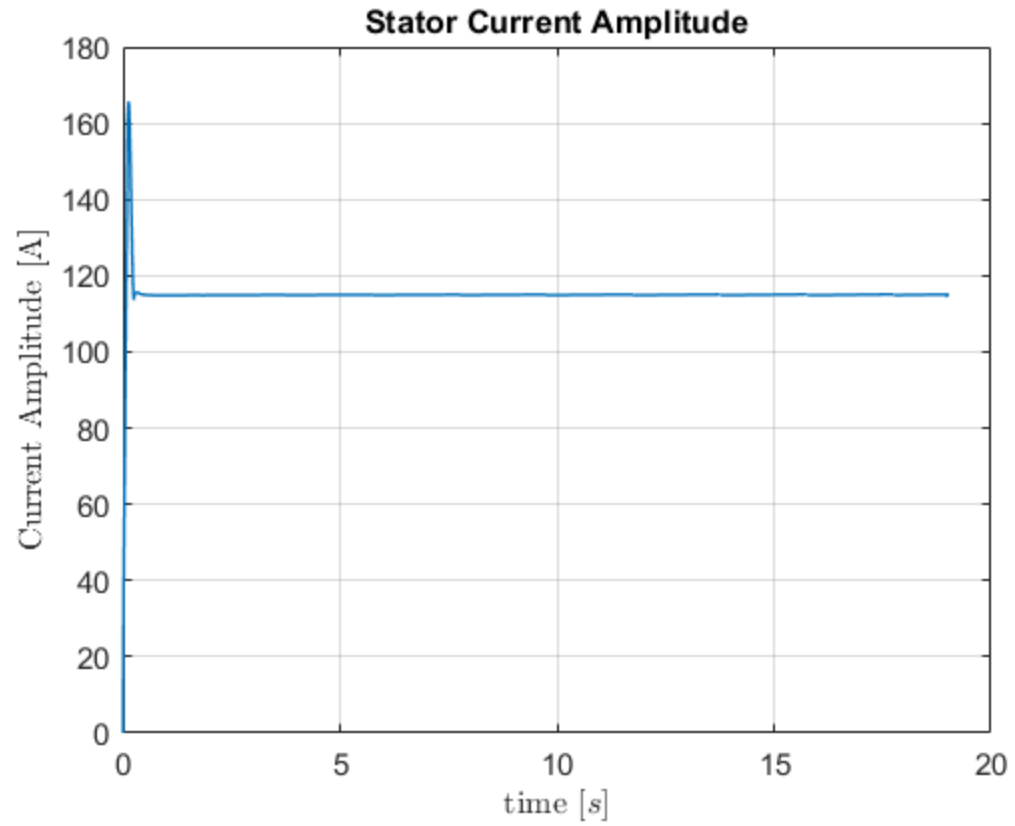
Considering that the efficiency of the gearbox is 100%, and the gearbox ratio is (1:8), one can see that the wheel torque is 8 times the motor torque. This allows the induction motor to work in higher rotation speeds, where it has better performance.

As one can see from the plot above, the wheels torque follows the 150 [Nm] mark, showing that the motor behaves as expected, given the fact that this reference was generated from a car dynamics system that had a constant 150 [Nm] wheel torque input.

For the motor current, the following results are obtained:

```
figure()
plot(track1_part2.time, abs(track1_part2.stator_current), 'LineWidth',
     1);
set( gca, 'FontSize', 11);
grid on;
title('Stator Current Amplitude');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Current Amplitude [A]', 'Interpreter', 'latex');

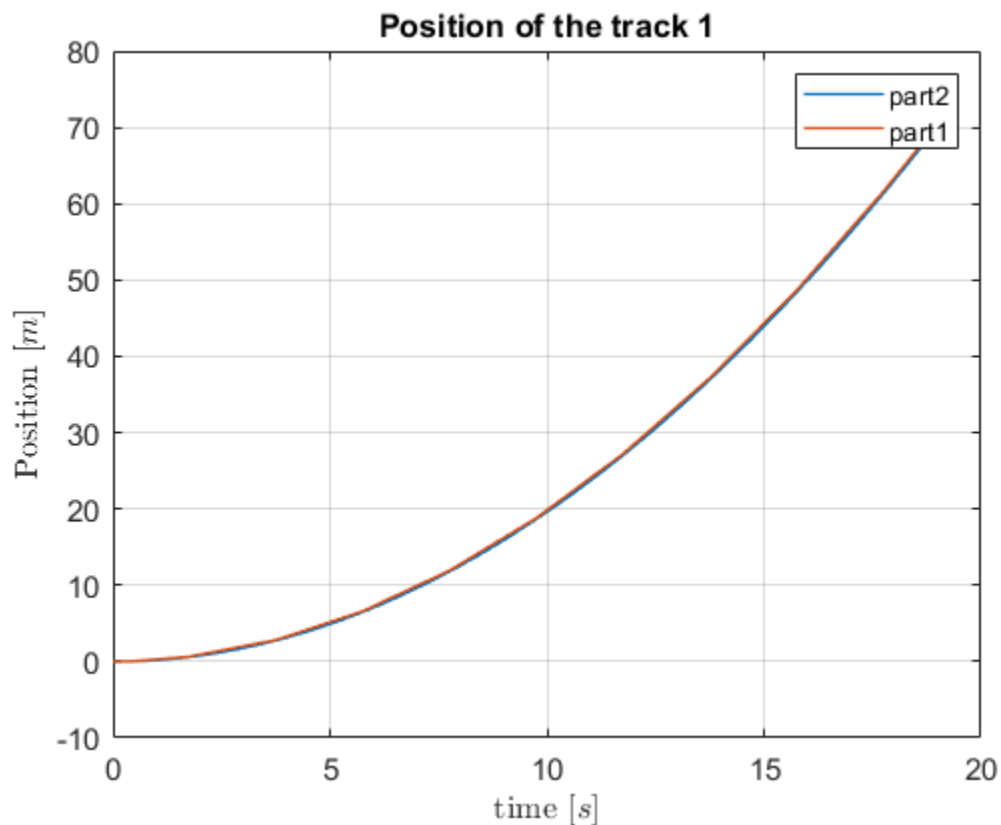
figure()
plot(track1_part2.time,
     angle(track1_part2.stator_current)*100, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Stator Current Angle');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Current Angle [\deg]', 'Interpreter', 'latex');
```



One can see that the stator current increases very rapidly in amplitude and then stabilises in around 115 Amperes. The amplitude -93.36°.

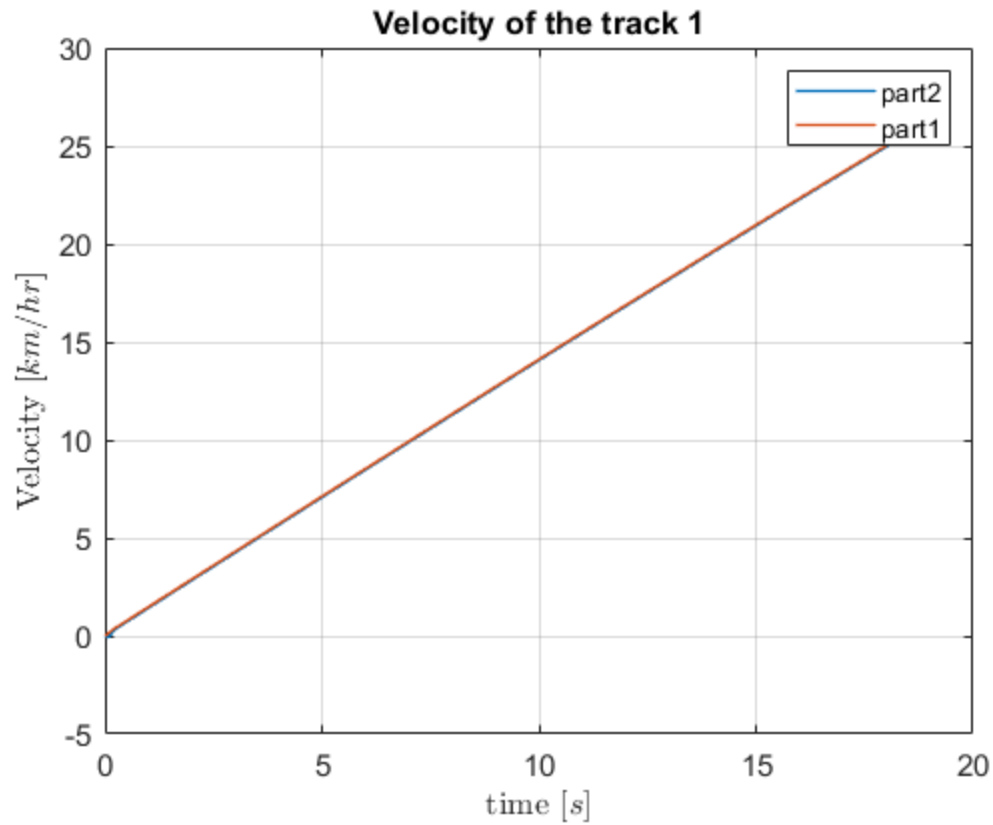
The position and velocity are:

```
track1.position(1) = [];  
track1.velocity(1) = [];  
figure()  
plot(track1_part2.time, track1_part2.position, 'LineWidth', 1);  
hold all  
plot(track1.time, track1.position, 'LineWidth', 1);  
set( gca, 'FontSize', 11);  
grid on;  
title('Position of the track 1');  
xlabel('time [s]', 'Interpreter', 'latex');  
ylabel('Position [m]', 'Interpreter', 'latex');  
legend('part2', 'part1');
```



```
figure()  
plot(track1_part2.time, track1_part2.velocity*3.6, 'LineWidth', 1);  
hold all  
plot(track1.time, track1.velocity*3.6, 'LineWidth', 1);  
set( gca, 'FontSize', 11);  
grid on;  
title('Velocity of the track 1');  
xlabel('time [s]', 'Interpreter', 'latex');  
ylabel('Velocity [km/hr]', 'Interpreter', 'latex');
```

```
legend('part2', 'part1');
```

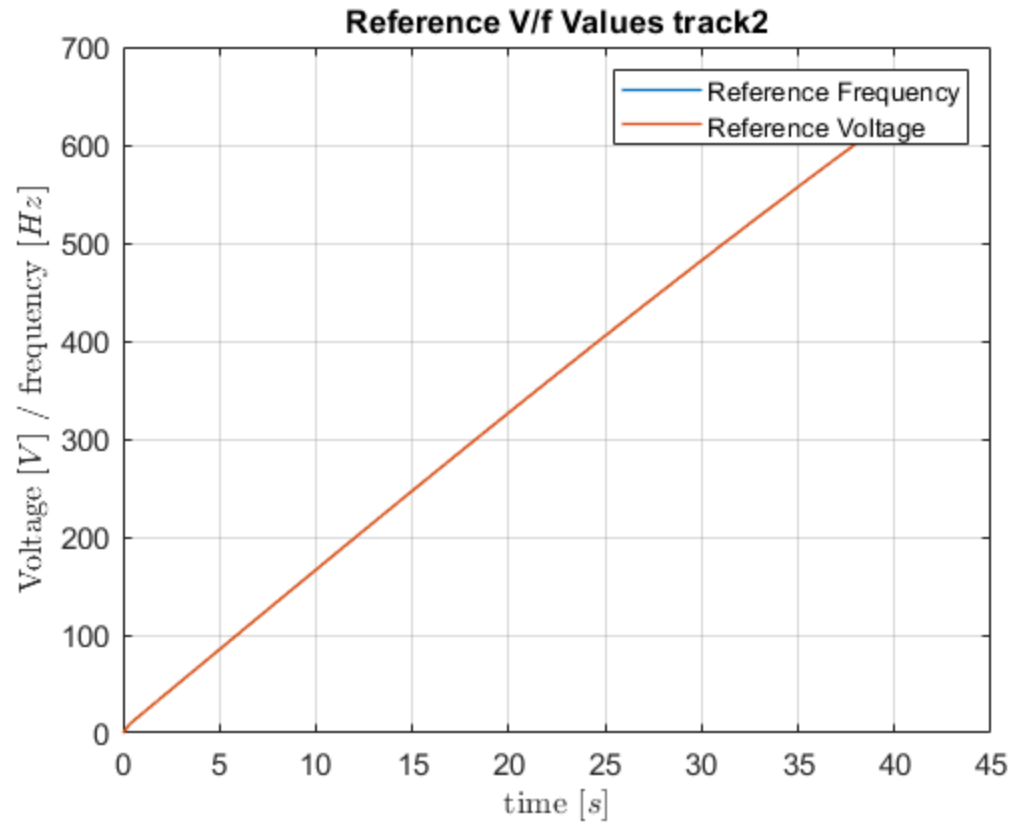


As expected, the positions and velocities for both parts are approximately the same.

```
slope_of_track = 0;  
reference_speed = reference_speed_track2;  
simulation_length_track = track2_length;  
track2_part2 = sim('vienna_car_part2');
```

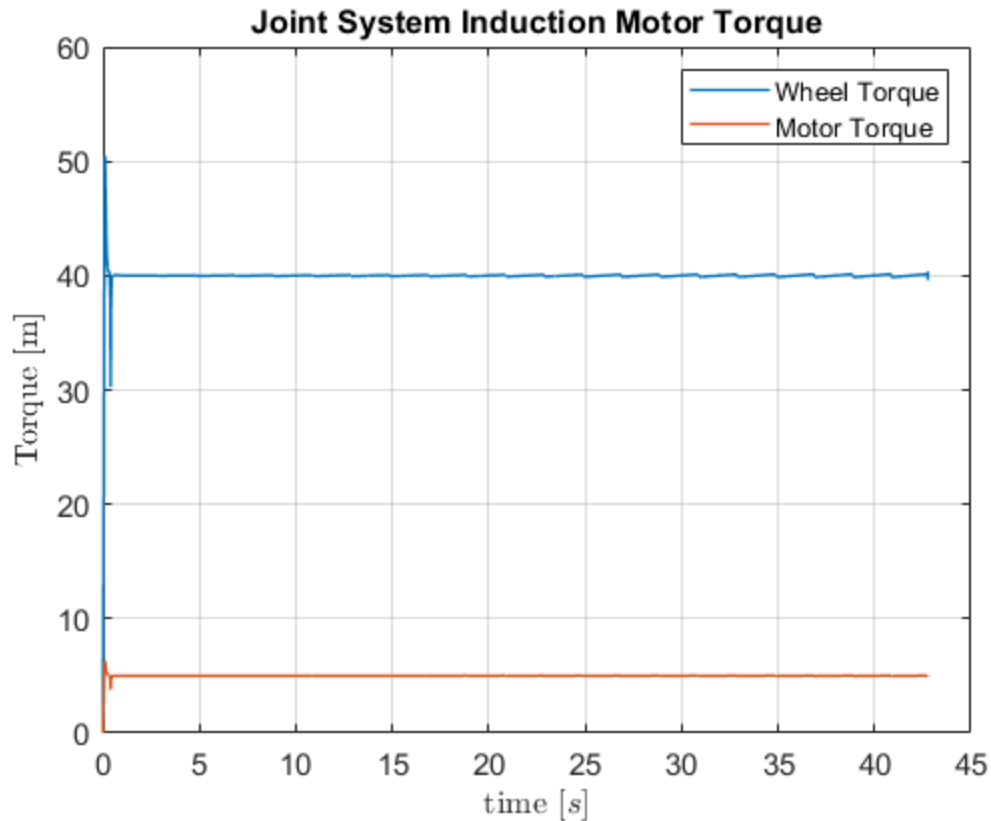
The reference V/f value for this track is given by:

```
figure()  
plot(track2_part2.time, track2_part2.reference_frequency, 'LineWidth',  
      1);  
hold all  
plot(track2_part2.time, track2_part2.reference_voltage, 'LineWidth',  
      1);  
set(gca, 'FontSize', 11);  
grid on;  
title('Reference V/f Values track2');  
xlabel('time [s]', 'Interpreter', 'latex');  
ylabel('Voltage [V] / frequency [Hz]', 'Interpreter', 'latex');  
legend('Reference Frequency', 'Reference Voltage')
```



The resultant wheel torque is given by:

```
figure()
plot(track2_part2.time, track2_part2.wheel_torque, 'LineWidth', 1);
hold all;
plot(track2_part2.time, track2_part2.motor_torque, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Joint System Induction Motor Torque');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Torque [m]', 'Interpreter', 'latex');
legend('Wheel Torque', 'Motor Torque');
```

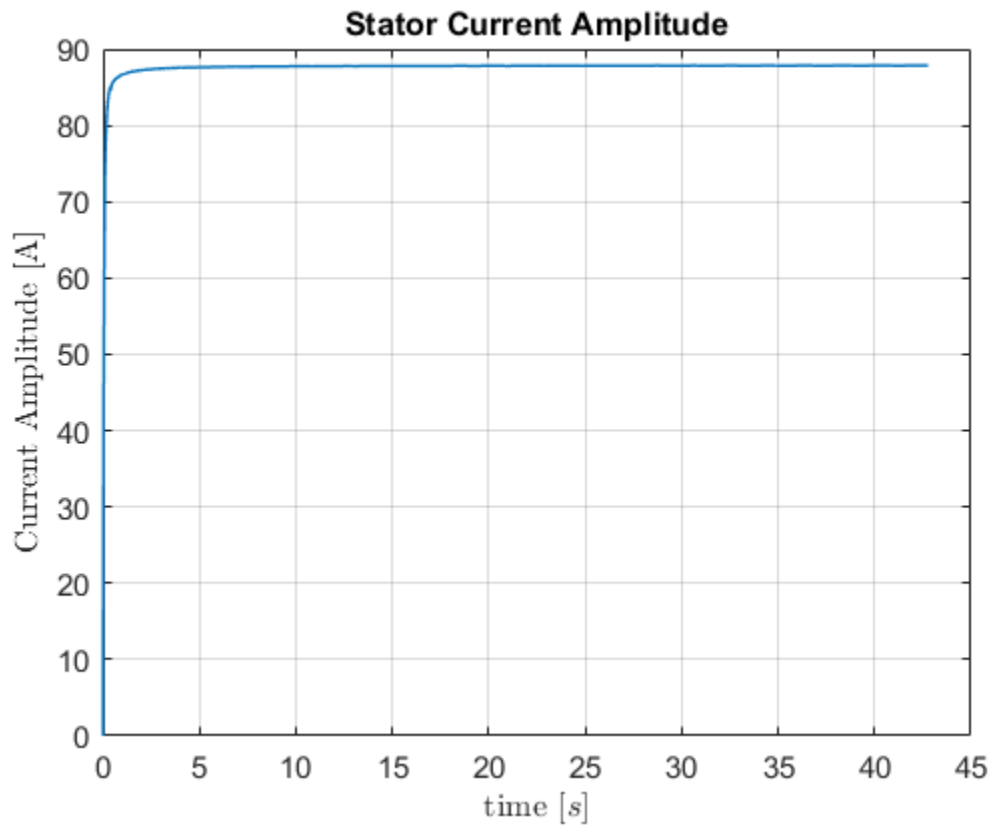
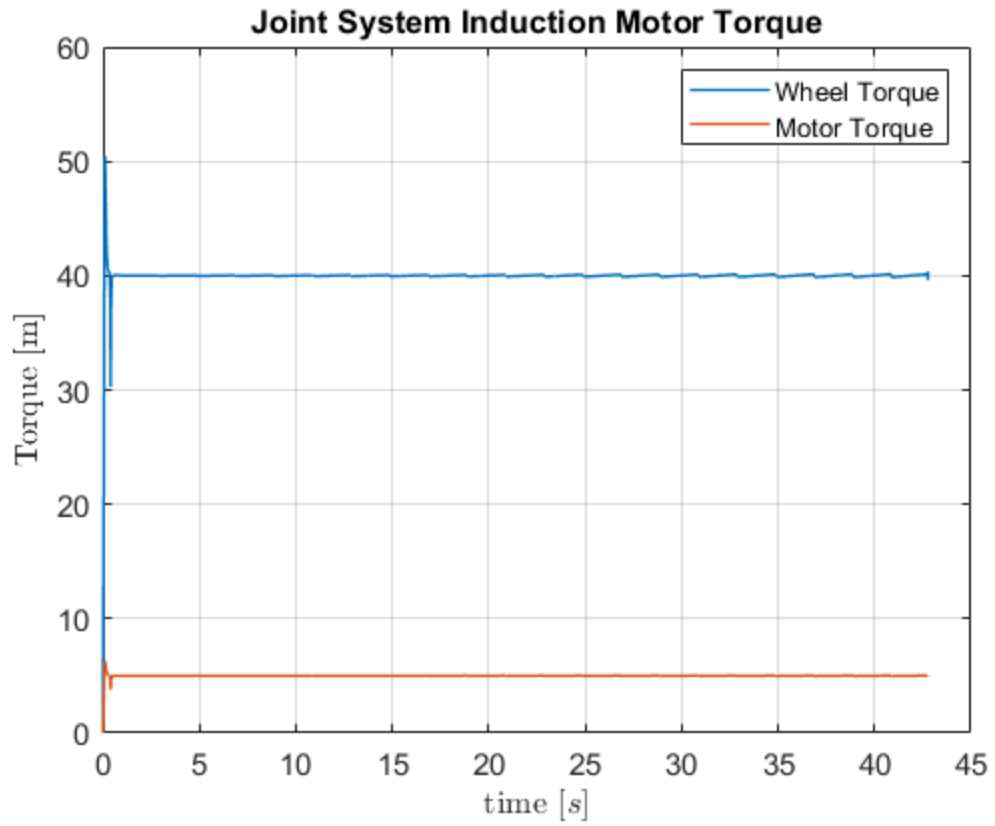
Considering that the efficiency of the gearbox is 100%, and the gearbox ratio is (1:8), one can see that the wheel torque is 8 times the motor torque. This allows the induction motor to work in higher rotation speeds, where it has better performance.

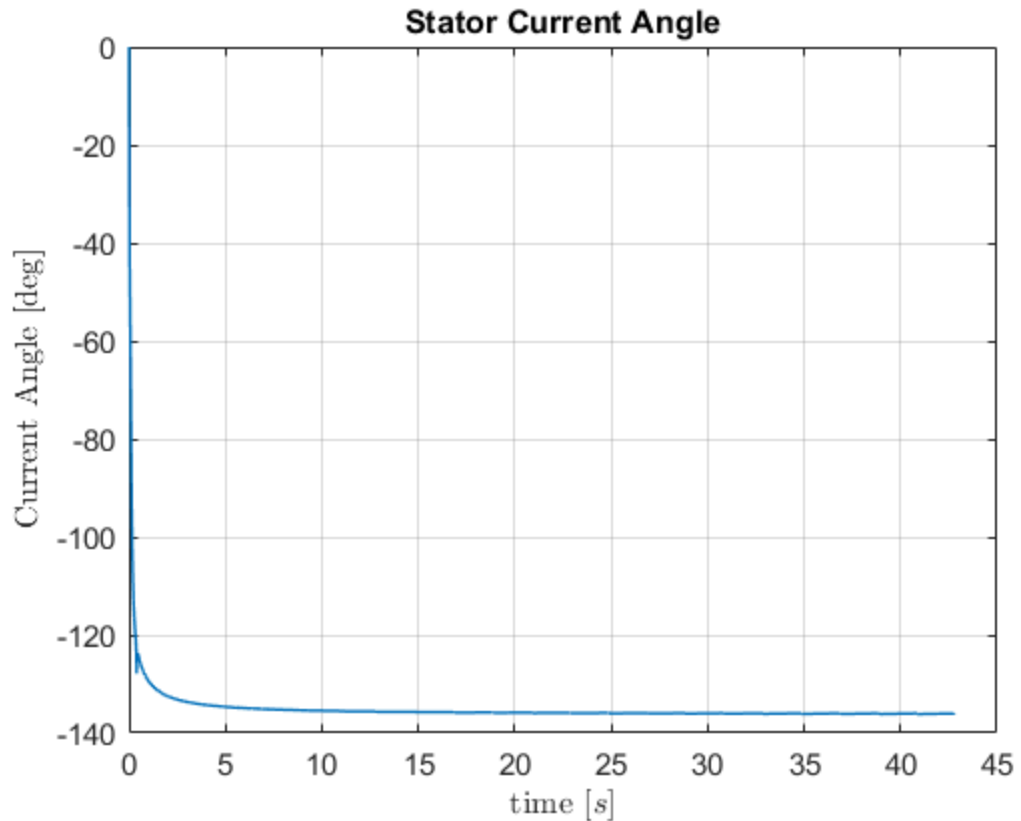
As one can see from the plot above, the wheels torque follows the 40 [Nm] mark, showing that the motor behaves as expected, given the fact that this reference was generated from a car dynamics system that had a constant 40 [Nm] wheel torque input.

For the motor current, the following results are obtained:

```
figure()
plot(track2_part2.time, abs(track2_part2.stator_current), 'LineWidth',
     1);
set( gca, 'FontSize', 11);
grid on;
title('Stator Current Amplitude');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Current Amplitude [A]', 'Interpreter', 'latex');

figure()
plot(track2_part2.time,
     angle(track2_part2.stator_current)*100, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Stator Current Angle');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Current Angle [\deg]', 'Interpreter', 'latex');
```

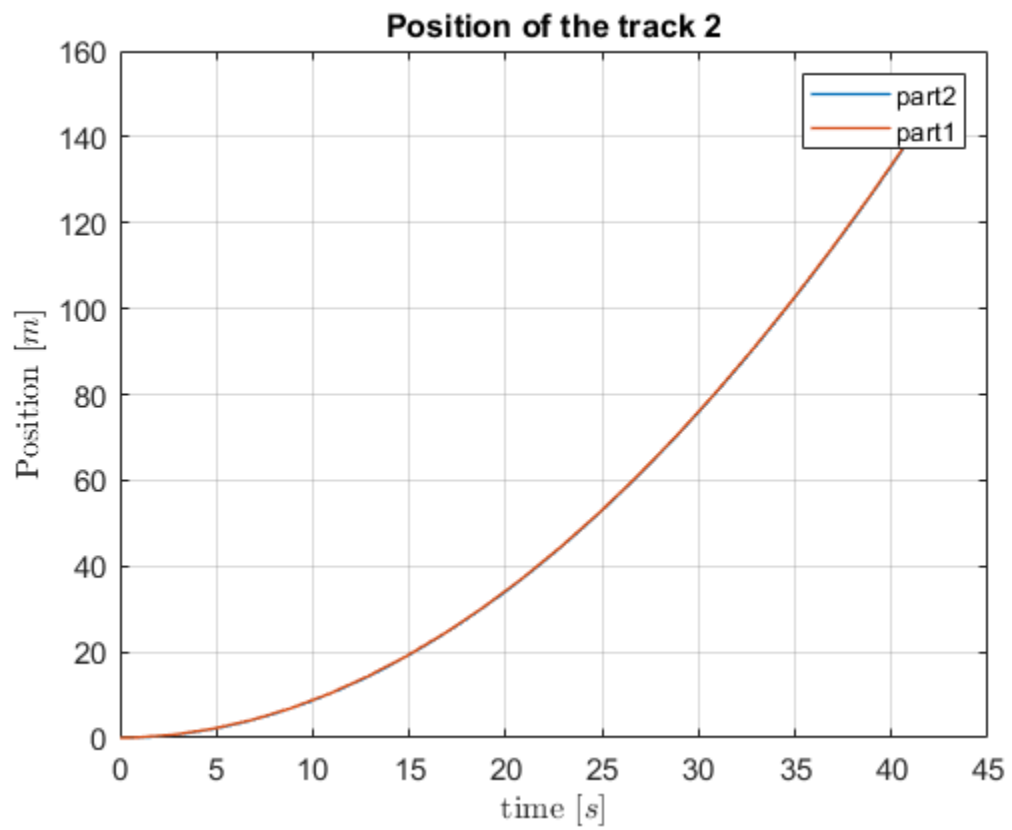
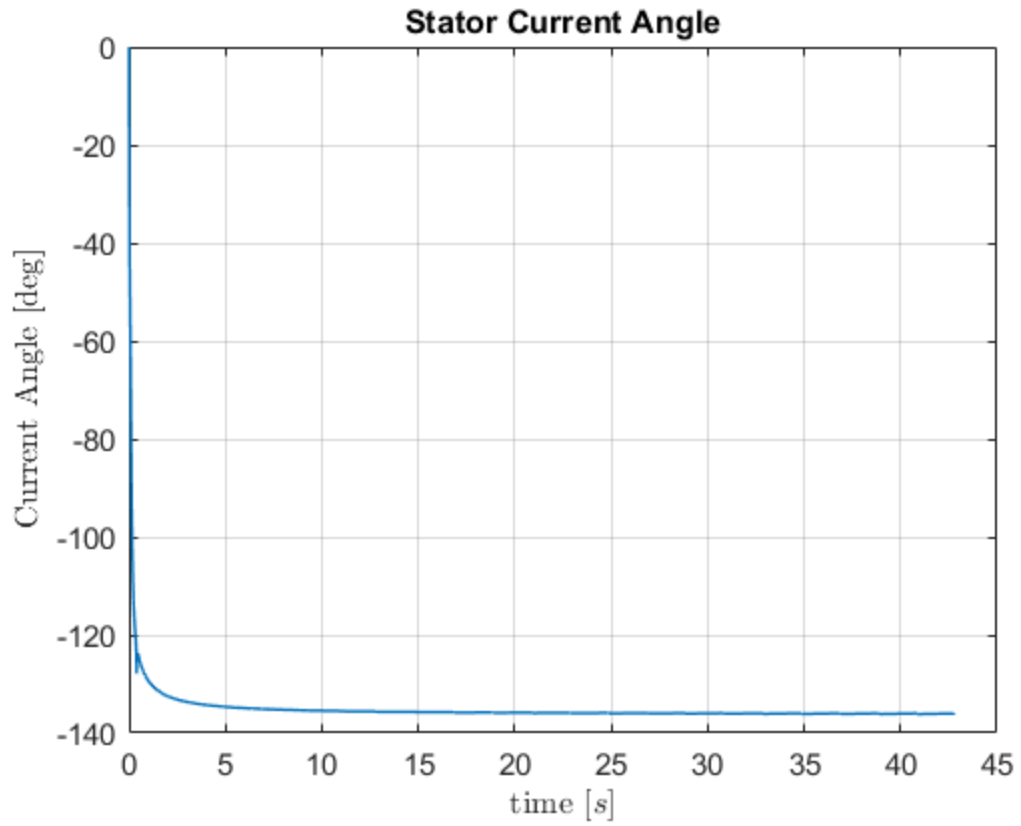




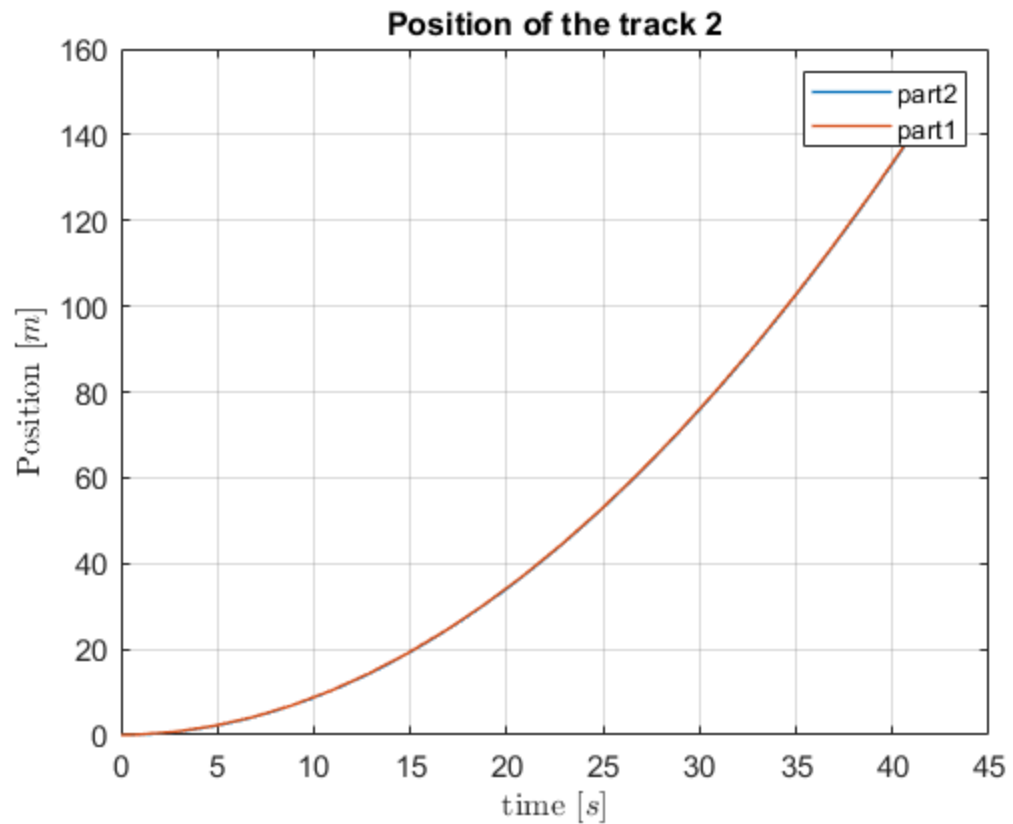
One can see that the stator current increases very rapidly in amplitude and then stabilises in around 87.8 Amperes. The amplitude -136.0°.

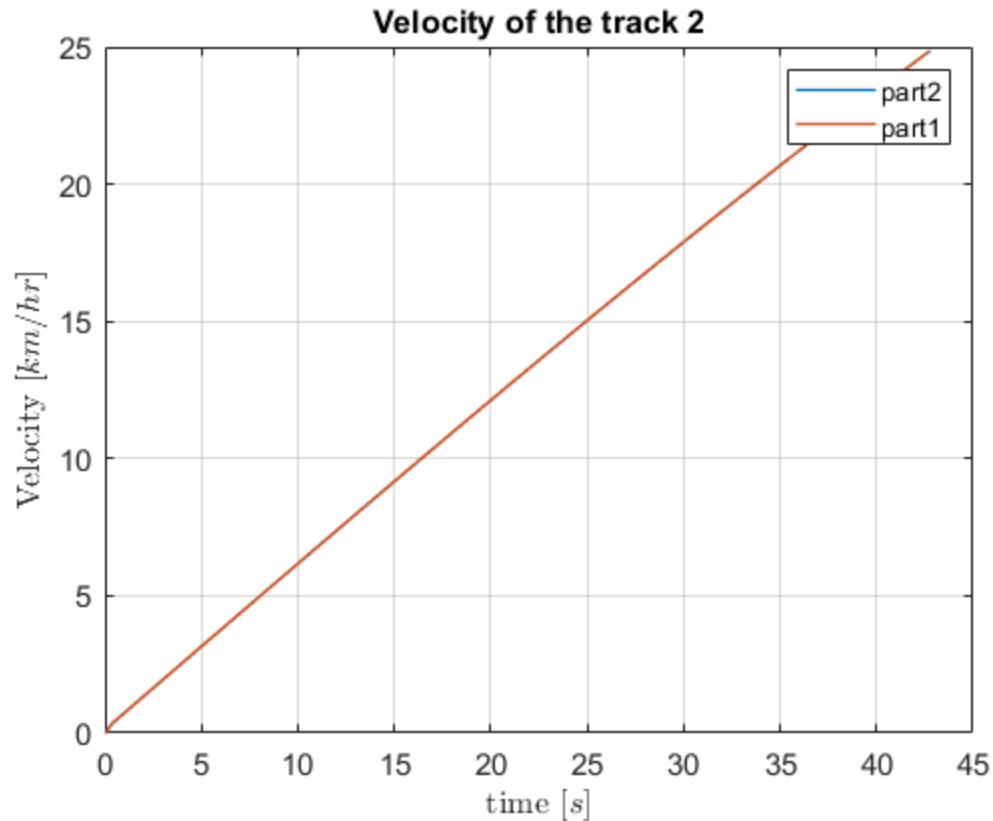
The position and velocity are:

```
track2.position(1) = [];  
track2.velocity(1) = [];  
figure()  
plot(track2_part2.time, track2_part2.position, 'LineWidth', 1);  
hold all  
plot(track2.time, track2.position, 'LineWidth', 1);  
set( gca, 'FontSize', 11);  
grid on;  
title('Position of the track 2');  
xlabel('time $[s]$', 'Interpreter', 'latex');  
ylabel('Position $[m]$', 'Interpreter', 'latex');  
legend('part2', 'part1');
```



```
figure()
plot(track2_part2.time, track2_part2.velocity*3.6, 'LineWidth', 1);
hold all
plot(track2.time, track2.velocity*3.6, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Velocity of the track 2');
xlabel('time  $[s]$ ', 'Interpreter', 'latex');
ylabel('Velocity  $[km/hr]$ ', 'Interpreter', 'latex');
legend('part2', 'part1');
```





Similarly to track 1, the velocity and the position are approximately the same as the ones in Vienna Project part 1.

Track 1 and Track 2 Comparisons

In terms of velocity and position, the induction motor follows, in both tracks, approximately the same results as in Vienna Project Part 1.

However, there is a big difference in the stator current:

- For track 1 the amplitude of the current is ~32% higher. This is due to the higher demand in power resultant from the slope in the track.
- In track 1, the angle of the current is -93° while in track2 it is $\sim 136^\circ$. The reason for this is that the higher power demand in track1 results in a greater reactive power needed for the magnetic coupling between the stator and rotor.

Conclusions

In this part of the project an equivalent circuit model of the induction motor along with a VFD were used to simulate the same tracks as in the first part of the project.

In general the same results were achieved as in the first part, having the car finishing the tracks at approximately the same time and speed.

Although this model presents good insights in the functioning of the car, the equivalent model is valid only for steady state, thus the transients are not appreciated in these simulations.

Published with MATLAB® R2019b