
Vienna Project with Speed Regulator

Table of Contents

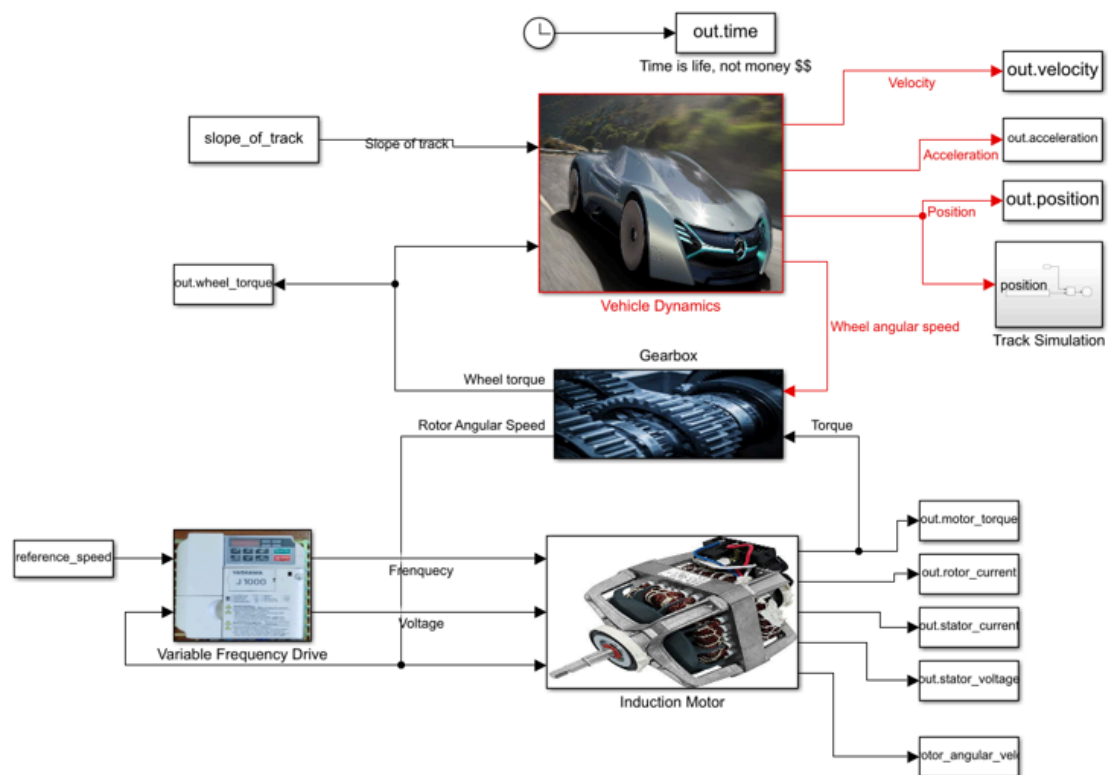
New Variable Speed Drive	1
Theretical approach of PID Control	2
Implementation	2
Proportional Control Track1	3
Proportional Control track2	7
Proportinal Integral Control	11
Proportional Control track2	15

New Variable Speed Drive

In this part of the vienna car project, the objective is to build upon the previous variable speed drive.

%The block diagram of the whole system is:

```
open_system('vienna_car_part3');
```



The objective of the variable frequency drive is to provide the induction motor with a stator voltage and frequency in order for the car to run. The variable frequency drive has to follow a reference value, in this

case, given by the rotational velocity of the wheels developed in the first part of the project. In order to follow the reference well, a PID control was added as shown in the following figure.

Theretical approach of PID Control

PID control stands for Proportional Integral Derivative control, in which the objective of having in the output a signal follows a reference given in the input. Each part of the PID has the following effect:

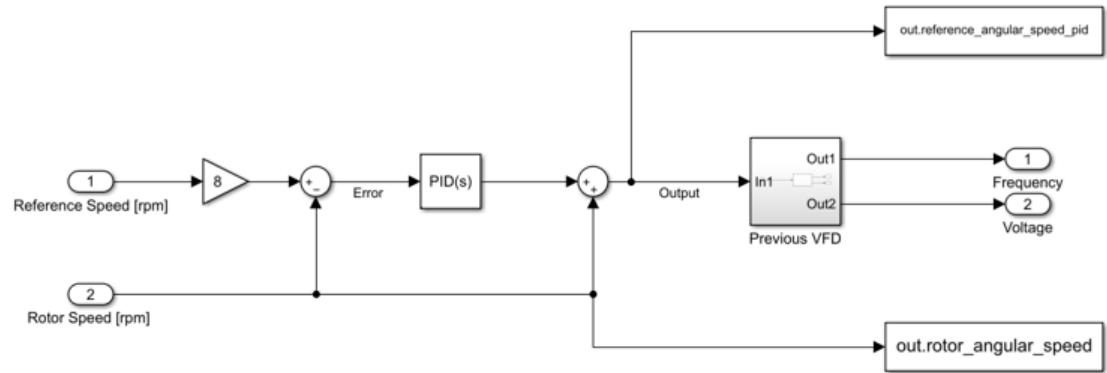
- Proportional- It is the biggest driver of the output to the reference, that is, the bigger the proportional constant, the faster the output will reach the reference. However, if the proportional component is too high, the output will overshoot the reference, representing the momentum of the system. trying to balance the previous effect, one might think that setting a proportional constant low will reduce overshoot and instability of the system, but if this constant is too low, the output will never reach the reference in real time.
- Differential - The differential component is responsible for measuring the rate of which the output is approaching the reference. The differential control then contributes with values that are contrary to the proportional control, mitigating the overshoot and allowing for higher response speeds. The downside is that the differential controller is sensitive to noise, and for an electric motor that has a very quick response, it may cause a lot of instability.
- Integral - The integral control has the effect of reducing the error between the output and the reference when the system is stabilising.

Implementation

The previous VFD is now complemented with the PID control:

1. It receives the reference speed (wheel speeds of the first part of the project), multiples it by 8 to create the desired motor speed.
2. It calculates the error between the reference speed and the actual speed of the motor.
3. The error enters in a ready made PID block from simulink.
4. The control signal of the PID block is then added with the same actual speed of the motors, resulting in the signal that will give the V/f values the exact same way as calculated in the second part of the project.

```
clc
clear
load(' ../vienna_car_part1/sim_part1 ');
close all
proportional_constat = 20;
integral_constant = 0;
open_system('vienna_car_part3/Variable Frequency Drive/Variabale
Frequency Drive Model');
```



Proportional Control Track1

```

slope_of_track = 3;
reference_speed = reference_speed_track1;
simulation_length_track = distance_A_B;
track1_part3 = sim('vienna_car_part3');

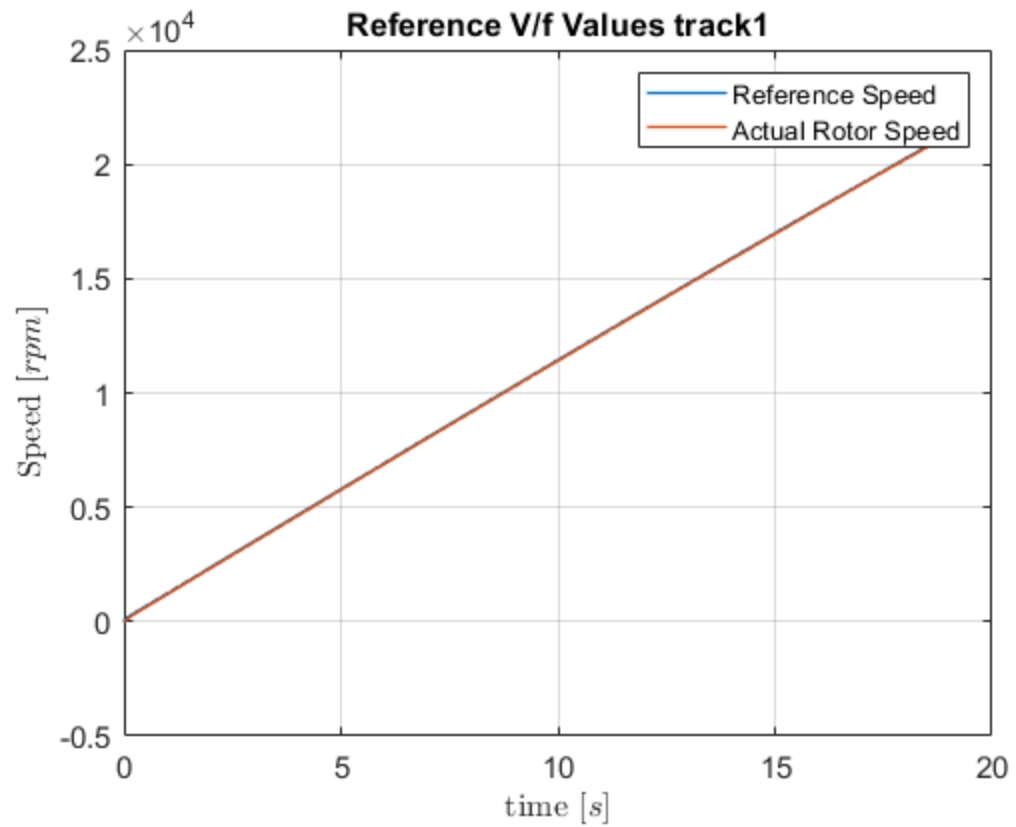
```

Reference speed

```

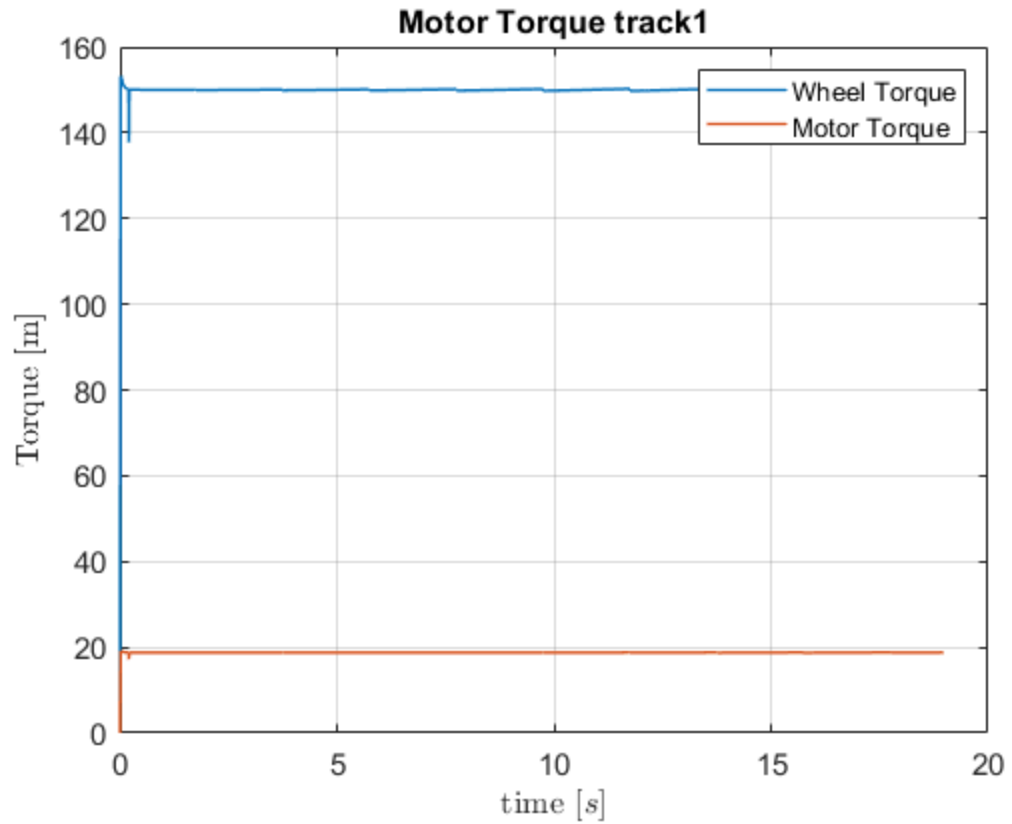
figure()
plot(track1_part3.time,
     track1_part3.reference_angular_speed_pid, 'LineWidth', 1);
hold all
plot(track1_part3.time, track1_part3.rotor_angular_speed, 'LineWidth',
     1);
set( gca, 'FontSize', 11);
grid on;
title('Reference V/f Values track1');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Speed [rpm]', 'Interpreter', 'latex');
legend('Reference Speed', 'Actual Rotor Speed');

```



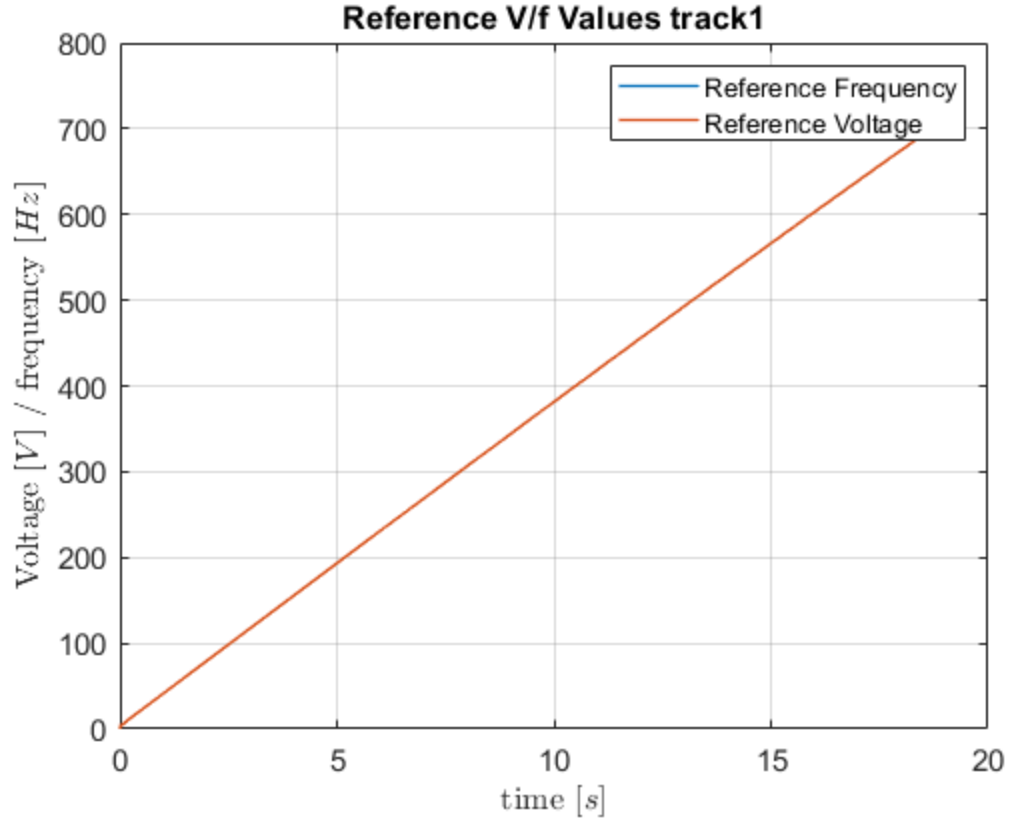
Motor Torque

```
figure()
plot(track1_part3.time, track1_part3.wheel_torque, 'LineWidth', 1);
hold all;
plot(track1_part3.time, track1_part3.motor_torque, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Motor Torque track1');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Torque [m]', 'Interpreter', 'latex');
legend('Wheel Torque', 'Motor Torque');
```



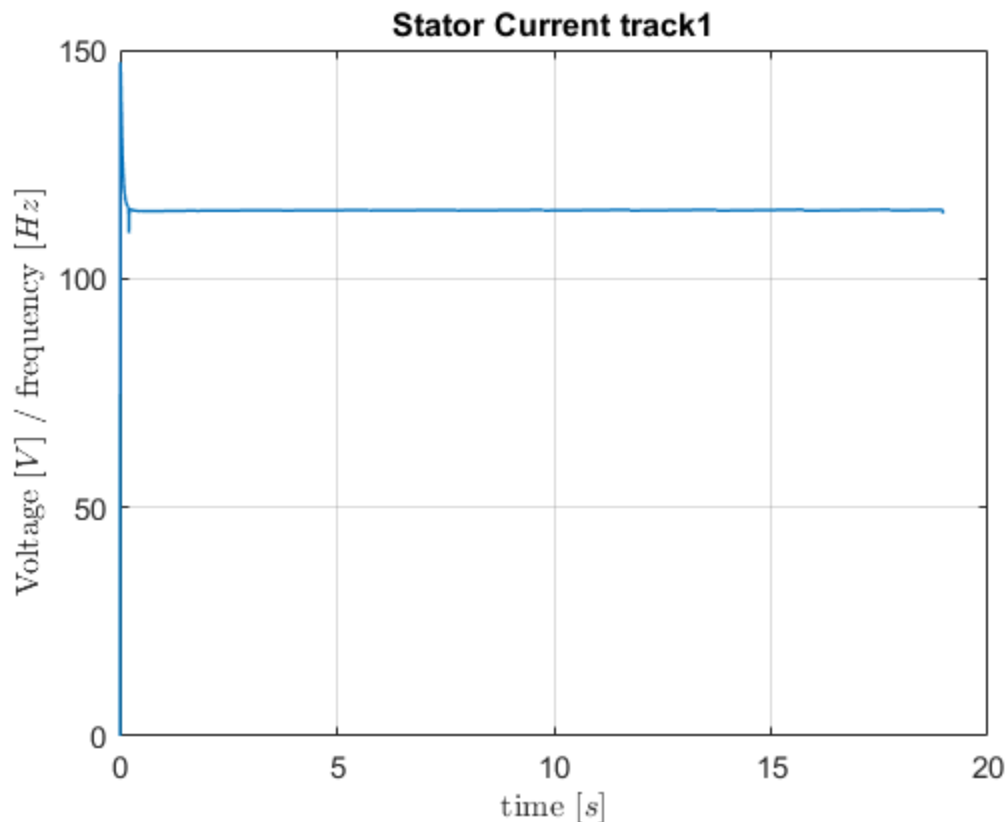
Voltage and frequency

```
figure()
plot(track1_part3.time, track1_part3.reference_frequency, 'LineWidth',
     1);
hold all
plot(track1_part3.time, track1_part3.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');
```



Stator Current

```
figure()
plot(track1_part3.time, abs(track1_part3.stator_current), 'LineWidth',
1);
set( gca, 'FontSize', 11);
grid on;
title('Stator Current track1');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Voltage [V] / frequency [Hz]', 'Interpreter', 'latex');
%% Position
% track1.position(1) = [];
% track1.velocity(1) = [];
% figure()
% plot(track1_part3.time, track1_part3.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('part3', 'part1');
```

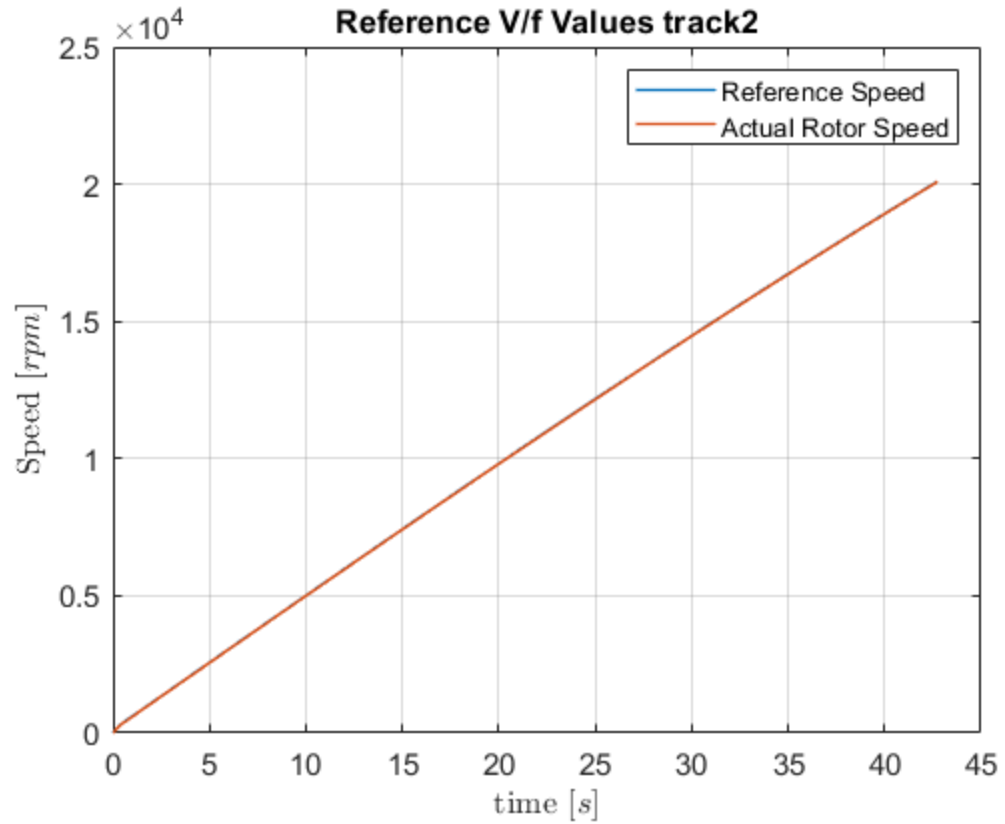


Proportional Control track2

```
slope_of_track = 0;
reference_speed = reference_speed_track2;
simulation_length_track = track2_length;
track2_part3 = sim('vienna_car_part3');
```

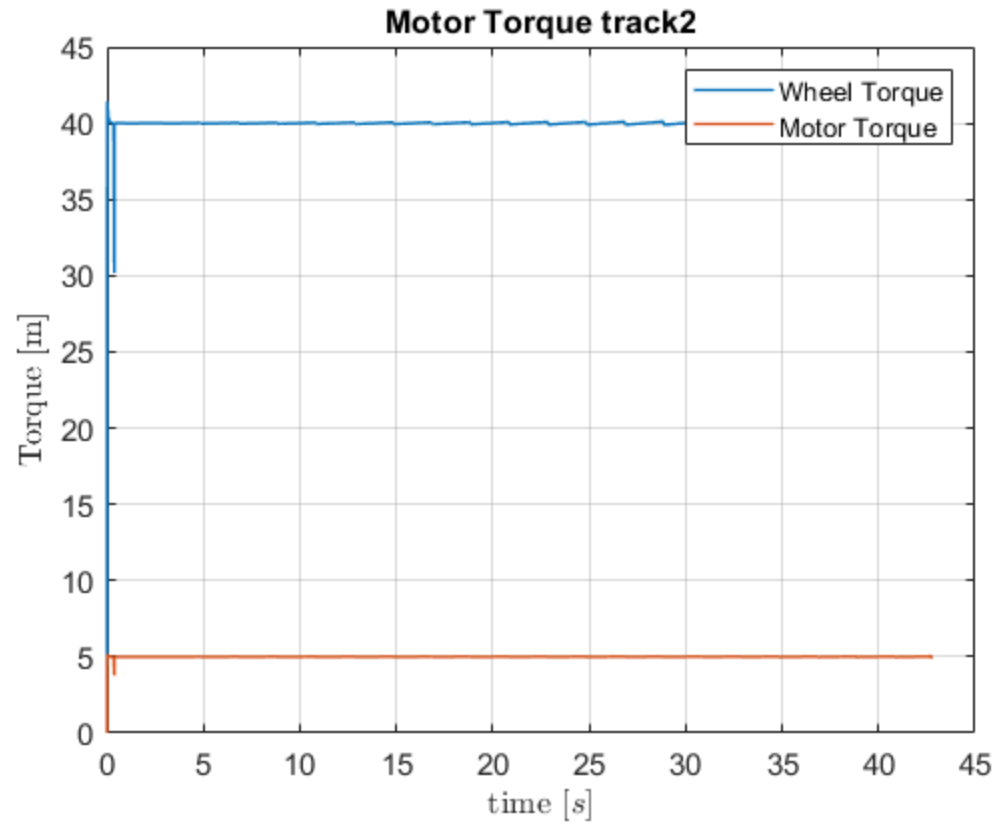
Reference speed

```
figure()
plot(track2_part3.time,
     track2_part3.reference_angular_speed_pid, 'LineWidth', 1);
hold all
plot(track2_part3.time, track2_part3.rotor_angular_speed, 'LineWidth',
     1);
set( gca, 'FontSize', 11);
grid on;
title('Reference V/f Values track2');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Speed [rpm]', 'Interpreter', 'latex');
legend('Reference Speed', 'Actual Rotor Speed');
```



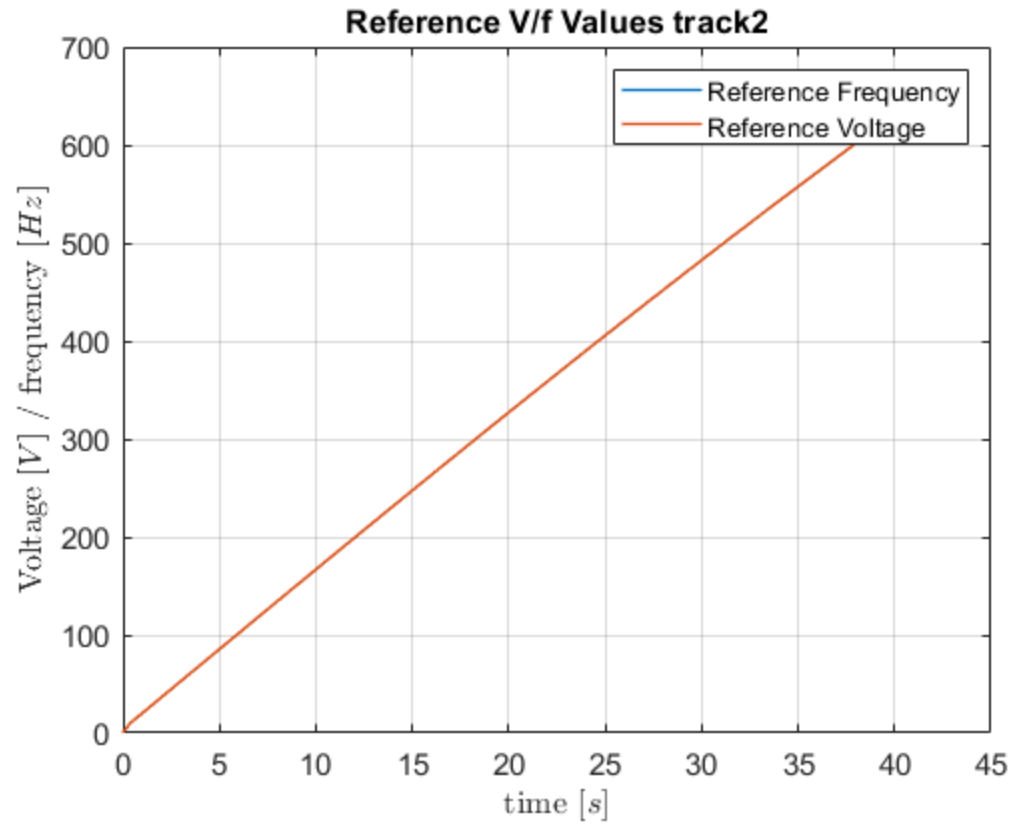
Motor Torque

```
figure()
plot(track2_part3.time, track2_part3.wheel_torque, 'LineWidth', 1);
hold all;
plot(track2_part3.time, track2_part3.motor_torque, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Motor Torque track2');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Torque [m]', 'Interpreter', 'latex');
legend('Wheel Torque', 'Motor Torque');
```

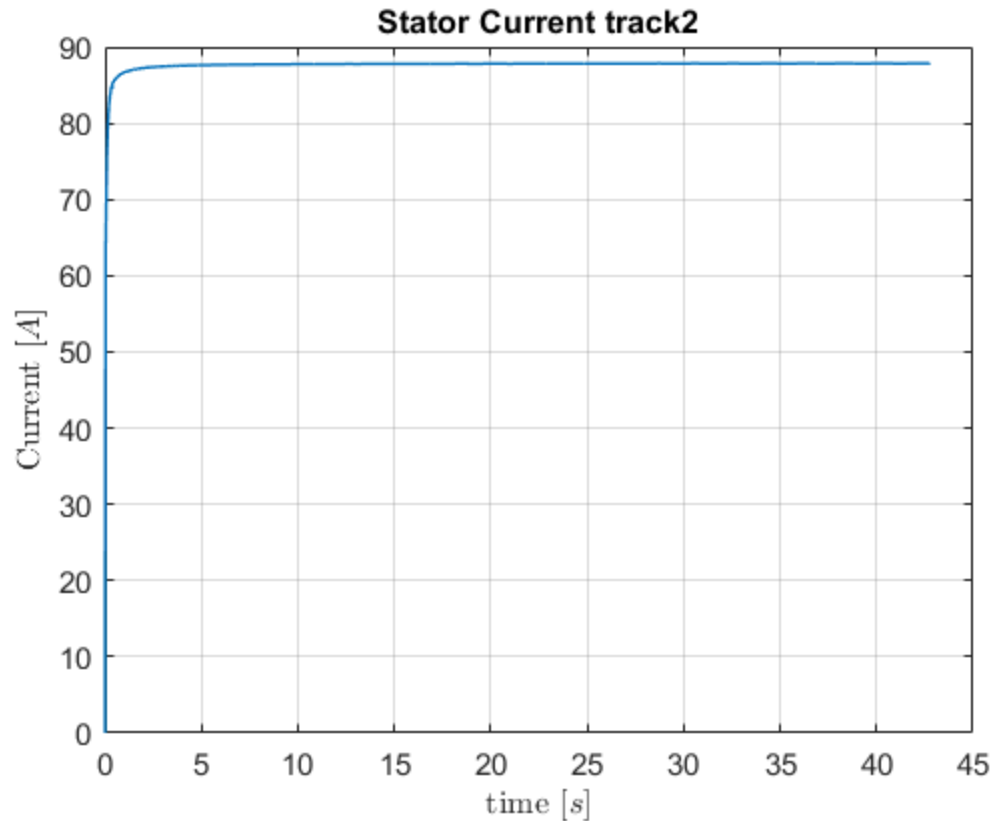
Voltage and frequency

```
figure()
plot(track2_part3.time, track2_part3.reference_frequency, 'LineWidth',
     1);
hold all
plot(track2_part3.time, track2_part3.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');
```



Stator Current

```
figure()
plot(track2_part3.time, abs(track2_part3.stator_current), 'LineWidth',
1);
set( gca, 'FontSize', 11);
grid on;
title('Stator Current track2');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Current [A]', 'Interpreter', 'latex');
%% Position
% track1.position(1) = [];
% track1.velocity(1) = [];
% figure()
% plot(track1_part3.time, track1_part3.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('part3', 'part1');
```

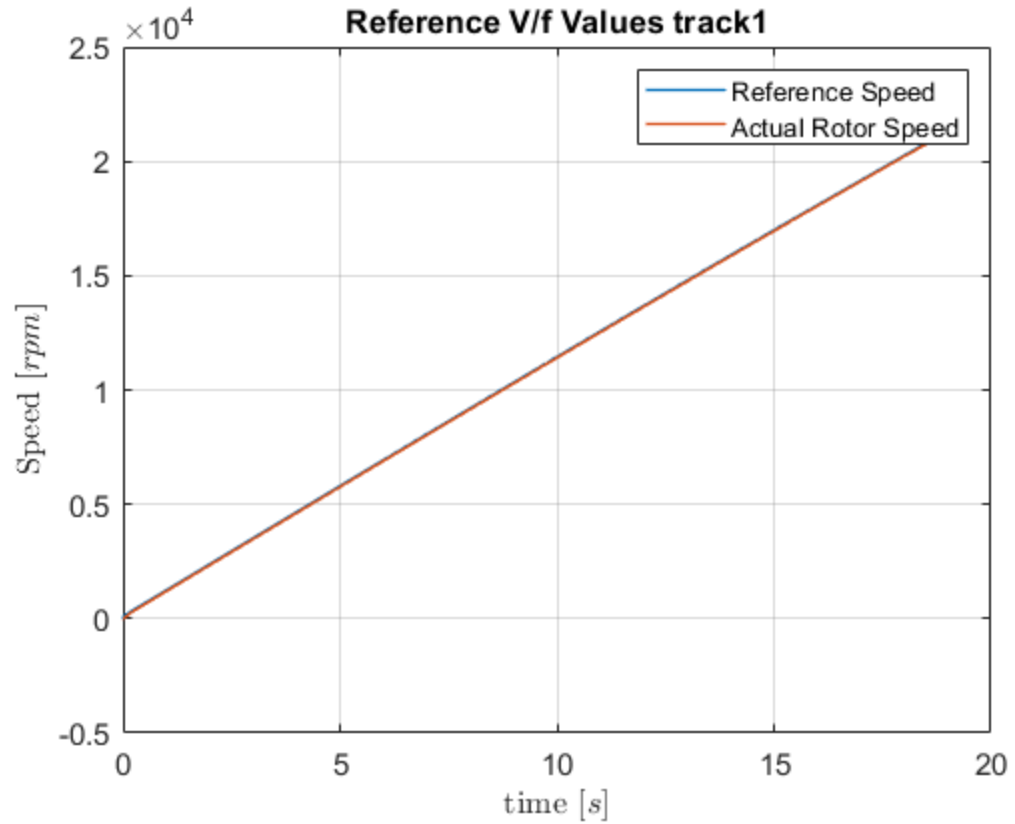


Proportional Integral Control

```
proportional_constat = 20;  
integral_constant = 200;  
slope_of_track = 3;  
reference_speed = reference_speed_track1;  
simulation_length_track = distance_A_B;  
track1_part3 = sim('vienna_car_part3');
```

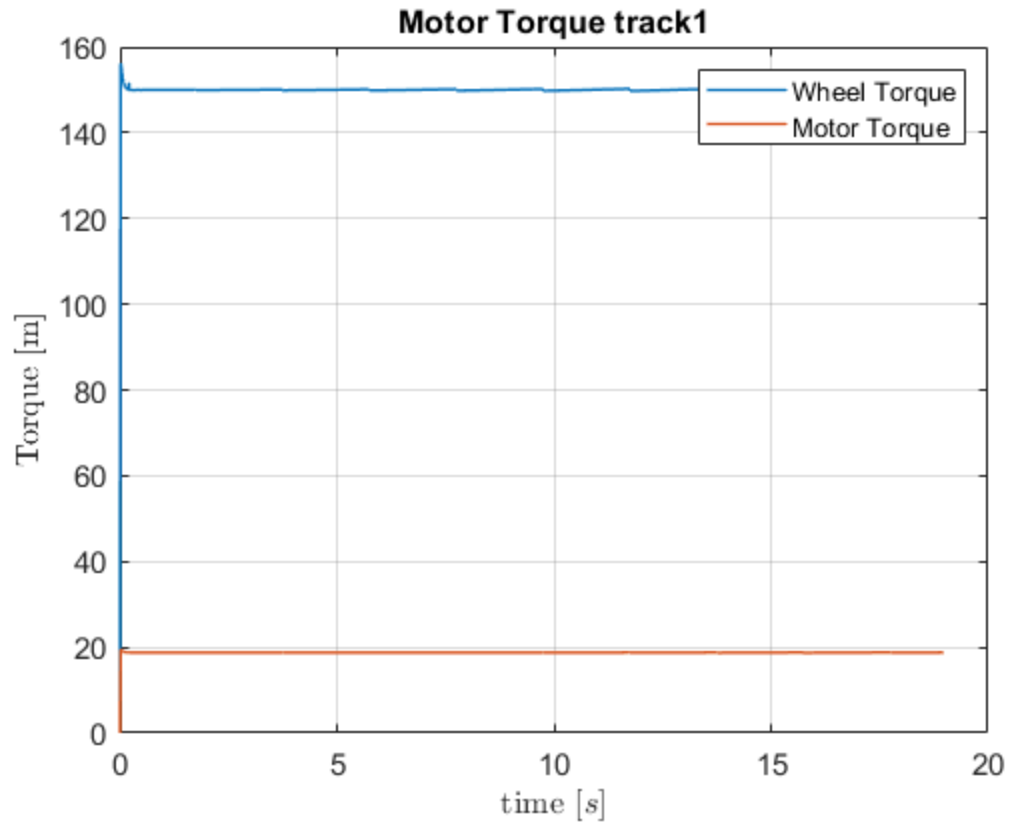
Reference speed

```
figure()  
plot(track1_part3.time,  
      track1_part3.reference_angular_speed_pid, 'LineWidth', 1);  
hold all  
plot(track1_part3.time, track1_part3.rotor_angular_speed, 'LineWidth',  
      1);  
set(gca, 'FontSize', 11);  
grid on;  
title('Reference V/f Values track1');  
xlabel('time [s]', 'Interpreter', 'latex');  
ylabel('Speed [rpm]', 'Interpreter', 'latex');  
legend('Reference Speed', 'Actual Rotor Speed');
```



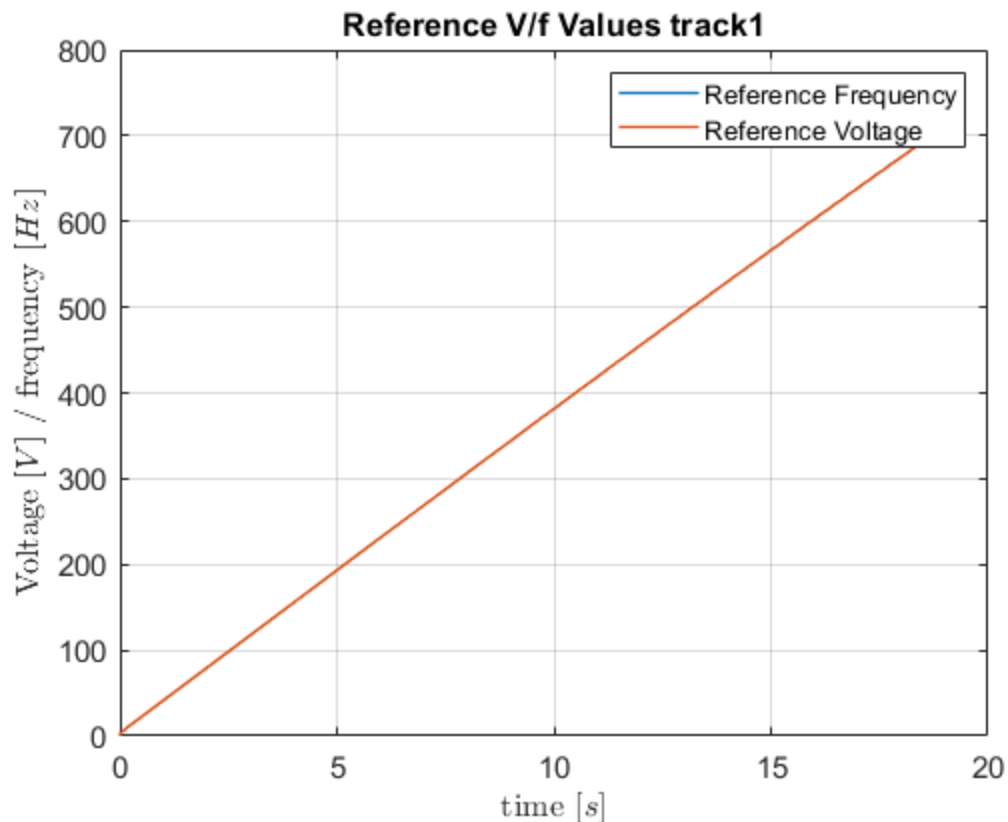
Motor Torque

```
figure()
plot(track1_part3.time, track1_part3.wheel_torque, 'LineWidth', 1);
hold all;
plot(track1_part3.time, track1_part3.motor_torque, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Motor Torque track1');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Torque [m]', 'Interpreter', 'latex');
legend('Wheel Torque', 'Motor Torque');
```



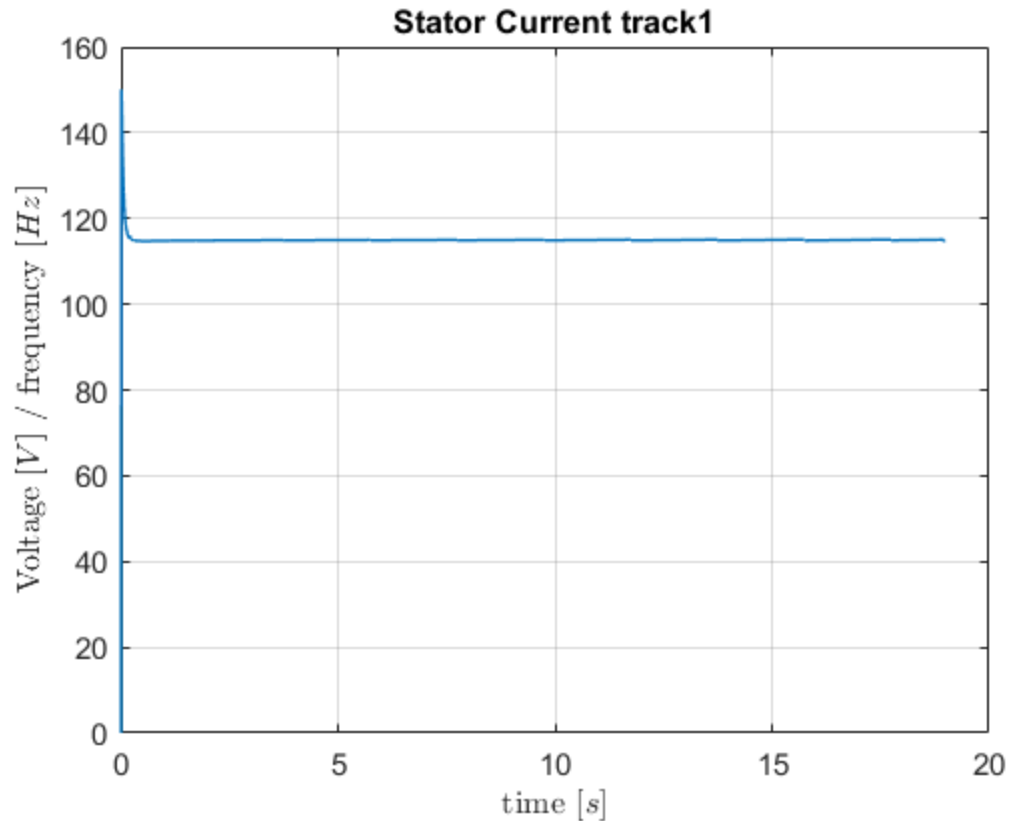
Voltage and frequency

```
figure()
plot(track1_part3.time, track1_part3.reference_frequency, 'LineWidth',
     1);
hold all
plot(track1_part3.time, track1_part3.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');
```



Stator Current

```
figure()
plot(track1_part3.time, abs(track1_part3.stator_current), 'LineWidth',
1);
set( gca, 'FontSize', 11);
grid on;
title('Stator Current track1');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Voltage [V] / frequency [Hz]', 'Interpreter', 'latex');
% %% Position
% track1.position(1) = [];
% track1.velocity(1) = [];
% figure()
% plot(track1_part3.time, track1_part3.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('part3', 'part1');
```

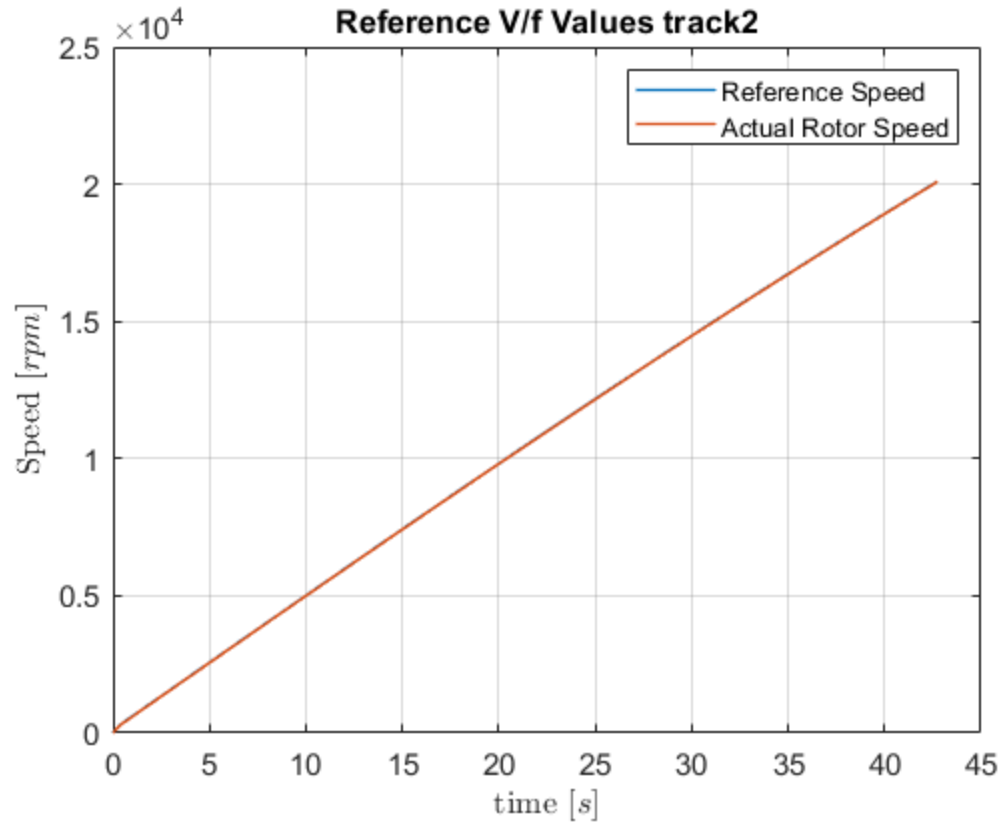


Proportional Control track2

```
slope_of_track = 0;
reference_speed = reference_speed_track2;
simulation_length_track = track2_length;
track2_part3 = sim('vienna_car_part3');
```

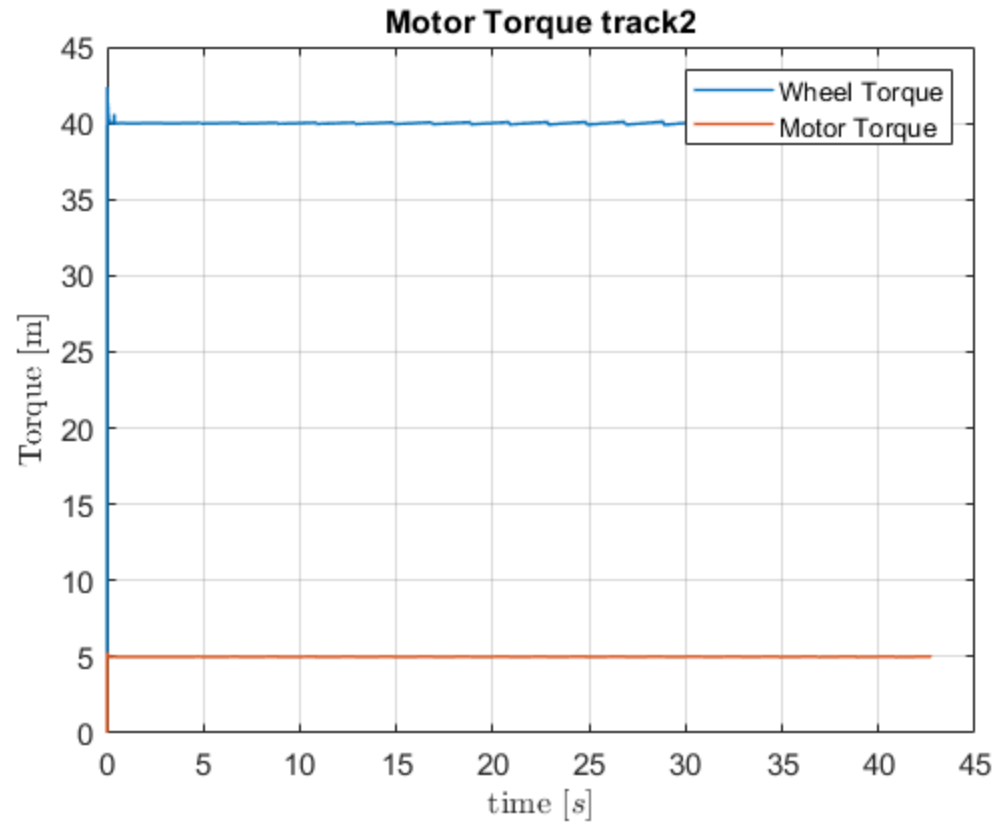
Reference speed

```
figure()
plot(track2_part3.time,
     track2_part3.reference_angular_speed_pid, 'LineWidth', 1);
hold all
plot(track2_part3.time, track2_part3.rotor_angular_speed, 'LineWidth',
     1);
set( gca, 'FontSize', 11);
grid on;
title('Reference V/f Values track2');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Speed [rpm]', 'Interpreter', 'latex');
legend('Reference Speed', 'Actual Rotor Speed');
```



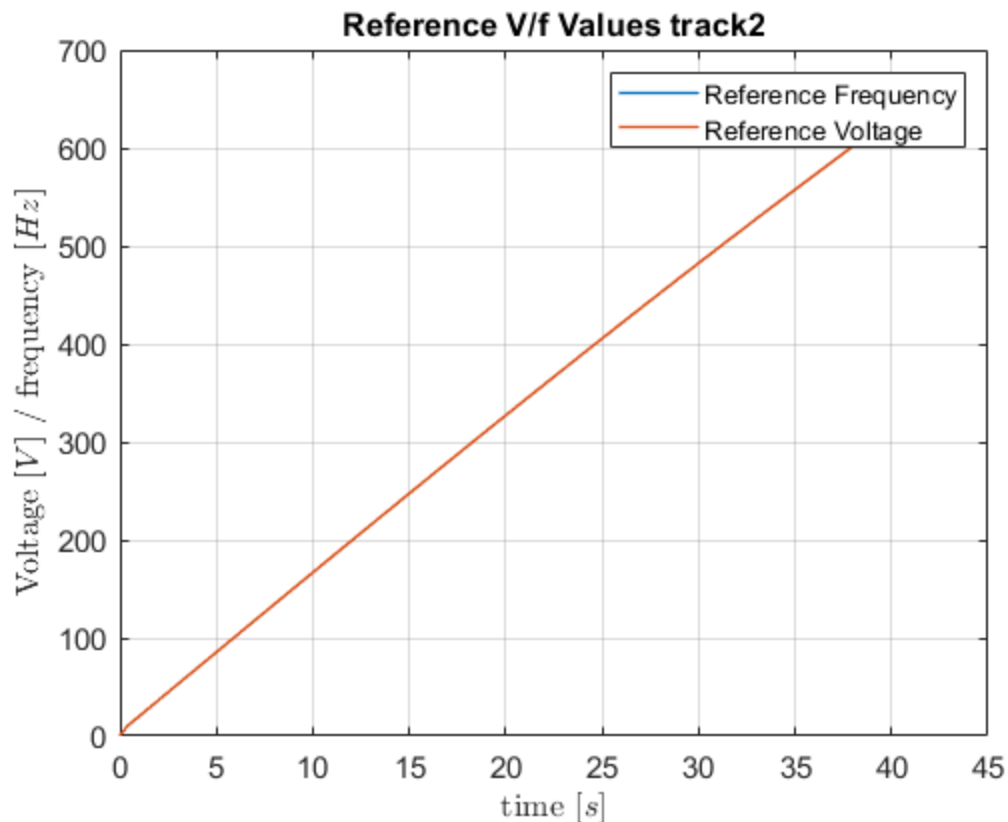
Motor Torque

```
figure()
plot(track2_part3.time, track2_part3.wheel_torque, 'LineWidth', 1);
hold all;
plot(track2_part3.time, track2_part3.motor_torque, 'LineWidth', 1);
set( gca, 'FontSize', 11);
grid on;
title('Motor Torque track2');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Torque [m]', 'Interpreter', 'latex');
legend('Wheel Torque', 'Motor Torque');
```

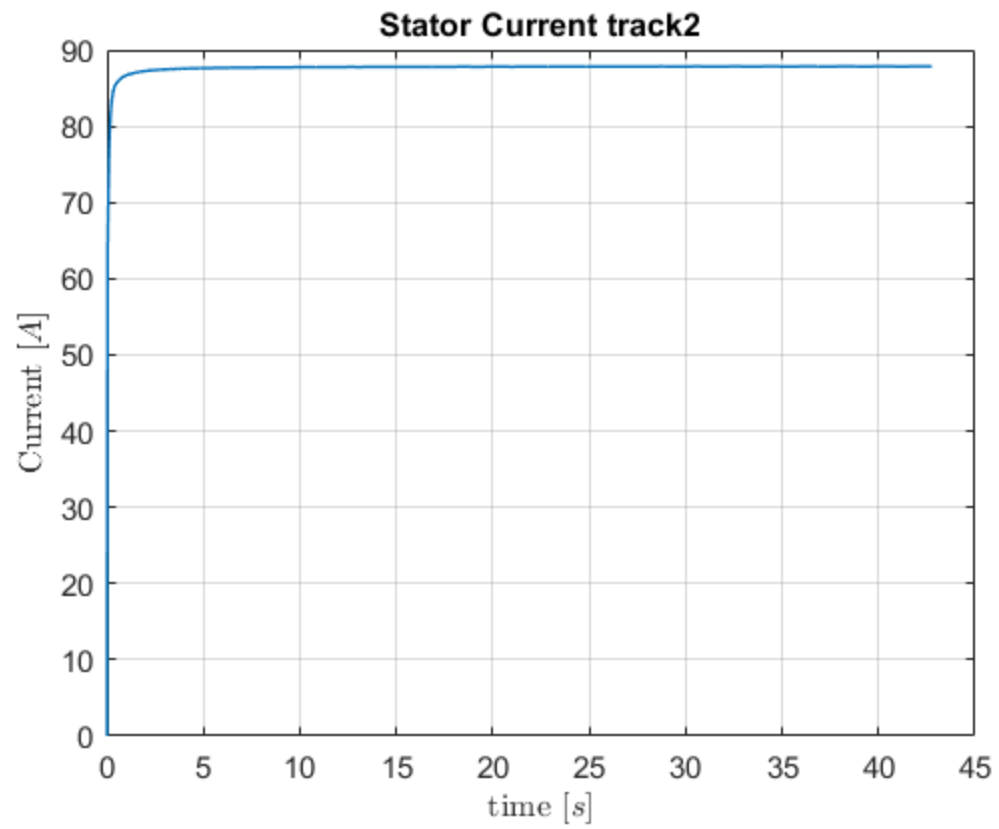
Voltage and frequency

```
figure()
plot(track2_part3.time, track2_part3.reference_frequency, 'LineWidth',
     1);
hold all
plot(track2_part3.time, track2_part3.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');
```



Stator Current

```
figure()
plot(track2_part3.time, abs(track2_part3.stator_current), 'LineWidth',
1);
set( gca, 'FontSize', 11);
grid on;
title('Stator Current track2');
xlabel('time [s]', 'Interpreter', 'latex');
ylabel('Current [A]', 'Interpreter', 'latex');
% %% Position
% track1.position(1) = [];
% track1.velocity(1) = [];
% figure()
% plot(track1_part3.time, track1_part3.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('part3', 'part1');
```



Published with MATLAB® R2019b