

Vehicles dynamics, planning and control of robotic cars (integration of
Master's thesis) [145581]



Vehicles dynamics, planning and control of robotic cars:

**Professor Gastone Pietro Rosati Papini &
Mattia Piccinini**

Full Name	ID	Study
Mario Tilocca	230394	EIT Autonomous Systems

Contents

1 Assignment 1	2
1.1 Exercise 1.1	2
1.2 Exercise 1.2	2
2 Assignment 2	3
2.1 Exercise 2.1	3
2.2 Exercise 2.2	5
3 Assignment 3	5
3.1 Exercise 3.1	5
4 Exercise 4	8
4.1 Scenario 1	8
4.2 Scenario 2	10
4.3 Scenario 3	13
5 Assignment 5	15
5.1 Exercise 5.1	15
5.1.1 Exercise 5.1.1	15
5.1.2 Exercise 5.1.2	16
5.2 Exercise 5.2	17
6 Assignment 6	19
6.1 Clothoid Controller	19
6.2 Arc Path Controller	21
6.3 Stanley Kinematic	24
6.4 Stanley Dynamic	26
7 Assignment 7	27
7.1 Exercise 7.1	27
7.2 Exercise 7.2	28
7.2.1 Arc-Path Controller	29
7.2.2 Clothoid controller	31
A Assignment 2 Matlab code	33
A.1 Exercise 2.1	33
A.2 Exercise 2.2	34
A.2.1 function resid_pure_Fx	35
A.2.2 function FXP	36
A.2.3 function resid_pure_Fx_varCamber	37
A.2.4 function resid_pure_Fx_varFz	37

1 Assignment 1

1.1 Exercise 1.1

Given the fact that in case of a pure longitudinal slip condition the force grows with the increase of the side slip κ , as it can be seen in Fig.[1] the force, F_{x0} suddenly increases when its value is approaching 0. For calculating the longitudinal force, the 2nd and 15th equation from Eq(20) from the Tire_Models.doc.pdf were followed. From the figure it can be observed that after reaching a peak of 2[kN] the force gradually decreases. Therefore when considering the optimal target value for the side slip, it should be in the range between [0, 0.15], thus when the force has reached or is about to reach its maximum value so that the longitudinal acceleration is consequently maximized.

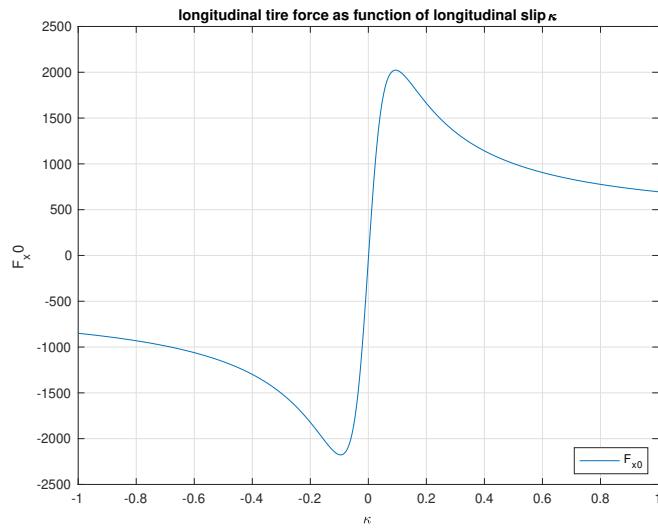


Figure 1: F_{x0} in pure longitudinal slip conditions as a function of slip κ in the given range

Given as initial condition a wheel rotational speed $= 70 \text{ rad/s}$, a tire effective rolling radius is $R_e = 0.2 \text{ m}$, a longitudinal component of tire contact point speed $v_{Cx} = 13 \text{ m/s}$, the computed longitudinal slip κ is equal to **0.0769**.

In case κ is greater than 0 the wheel is accelerating, while on the contrary if κ has a negative value the wheel is braking and if $\kappa = -1$ the wheel is locked. As the calculated value of the longitudinal slip is greater than 0, also in these conditions the wheel is accelerating and this is confirmed by the fact that the resulting longitudinal tire force is equal to **1990.7 N**, thus at the peak of the longitudinal tire force as for the case of Fig.[1].

For calculating the cornering stiffness C_{fk} , the following approach was followed. Taking the derivative of Eq(7) from the provided Tire_Models.doc.pdf and evaluating the obtained derivative for κ equal to 0 it was obtained a value of **47.89 kN/rad**. Furthermore from the theory we know that the linear approximation Pacejka curve is acceptable up to the point κ reaches unity values, thus in case it is -1 the wheel locked while in case it is 1 the wheel is spinning.

1.2 Exercise 1.2

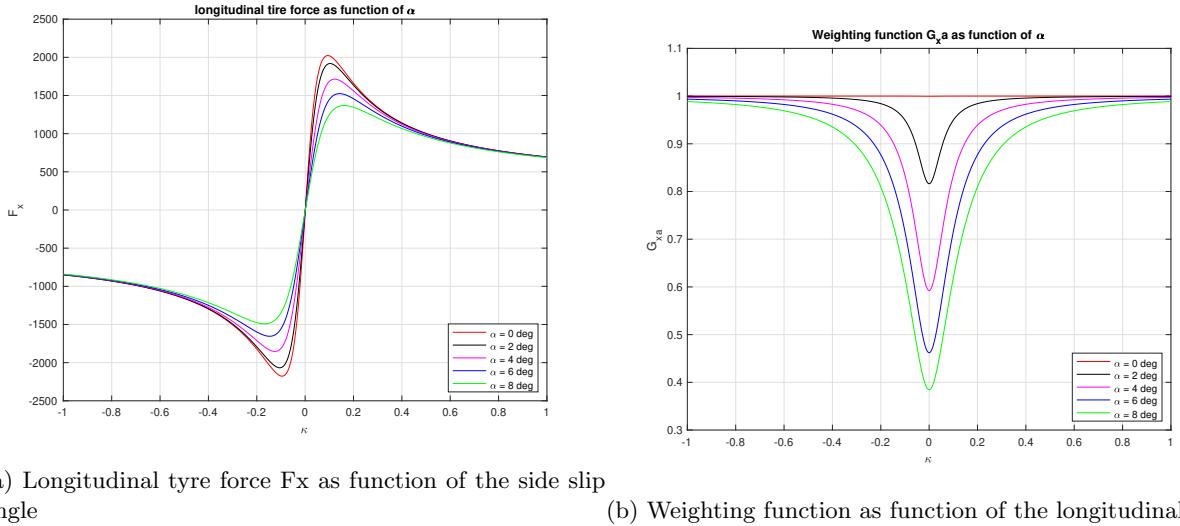
With the given parameters, the side slip angle α was calculated following the equations below:

$$\alpha = -\tan\left(\frac{V_{cy}}{V_{cx}}\right)$$

where V_{cy} and V_{cx} represent the contact point velocities for the lateral and longitudinal components. The obtained result gave $\alpha = \mathbf{0.0865}$.

The resulting combined tyre force which was calculated following the approach explained in the Eq(20) yielded a value of **2002.4**

Following the system of equation presented in Eq(20) the longitudinal tyre force and weighting function was calculated. As it can be seen in Fig[2a] the behavior of the longitudinal tyre force tends to have decreased peak values and a consequent less sharp shrink of the curve with an higher value of the side slip angle. This behavior can be addressed to the fact that the tire force is directly proportional to the weighting function as it shown in eq(1). The behavior of the weighting function is shown in Fig.[2b].



(a) Longitudinal tyre force F_x as function of the side slip angle

(b) Weighting function as function of the longitudinal slip

Figure 2: Illustration of various images

$$F_x = G_{x\alpha} \cdot F_{x0} \quad (1)$$

2 Assignment 2

2.1 Exercise 2.1

In Fig.[3] it is shown the raw data empirically acquired while running the tests. From the plots it can be observed as the longitudinal force and longitudinal slip are not particularly dependant on the variation of the other parameters such as pressure or the camber angle. The behaviors of the 5 parameters seem to be independent from one another. The empirical data was measured by varying four different loads for the vertical force F_z , three different values for the side slip angle α , two different values for the pressure and three values for the camber angle γ . Varying the above mentioned parameters to such extent allows to create a dataset which can evaluate the behavior of the tyres under a very different range of settings.

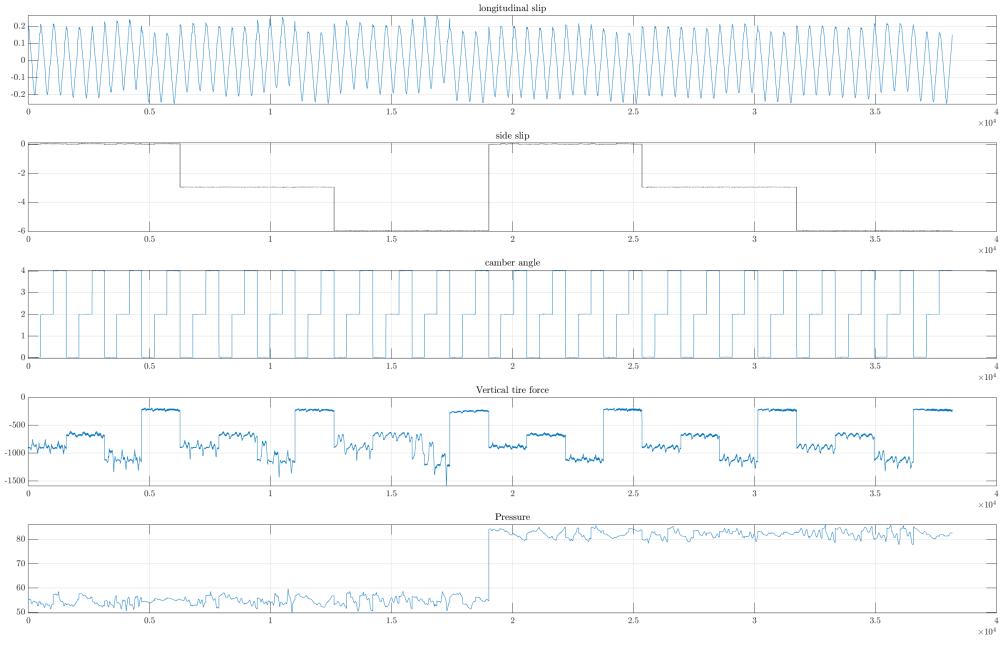


Figure 3: F_x behavior for the 3 different side slip angles α

In Fig[4a] it can be observed that the longitudinal force gradually increase as an higher load is applied as for the vertical force. Conversely in Fig.[4b] it is shown the behavior of the longitudinal force for the different values of the side slip angle. It can be observed that with an higher side slip the vehicle's longitudinal force decreases. This phenomenon can be explained as with an increase side slip the available gripping tyre surface might be decreased therefore the longitudinal force would decrease. In order to counter this, the vertical force F_z should be increased.

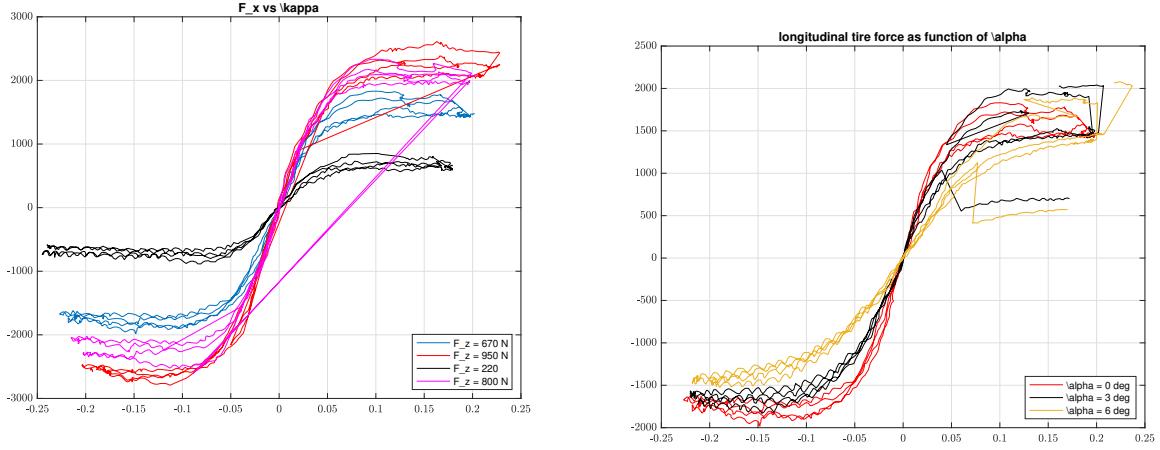


Figure 4: Illustration of various images

2.2 Exercise 2.2

For this task a fitting of the of the tyre data based on an optimization approach of the Pacejka magic formula was implemented. Despite several attempts the results were not as expected and only the first fitting was achieved as shown in the figure below. The developed code for the second assignment can be found in the appendix at the end of the document.

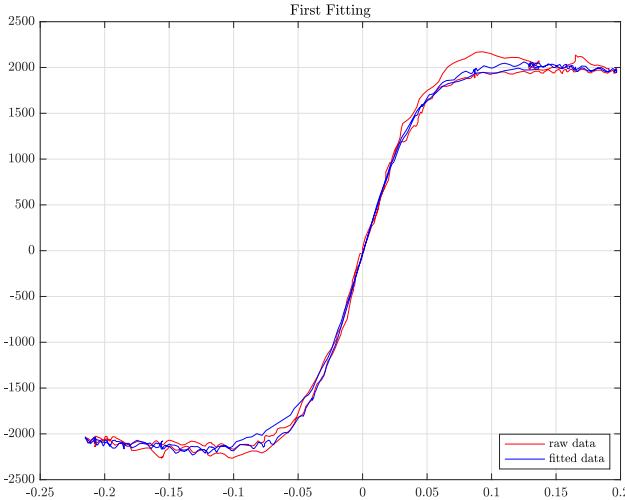


Figure 5: First fitting results for $F_{z0} = 890\text{N}$, $\gamma=0$ $\alpha=0$

3 Assignment 3

3.1 Exercise 3.1

From the data is shown in Fig[6a] it can be assumed that the car did not have straight trajectory, a fact which is confirmed as the data was acquired in a track where values constantly change.

Moreover the car was equipped with a Hall effect sensor, which is a common type of sensor used to measure the velocity through the wheel rotational speed. Each time the wheel makes an entire turn a corresponding voltage is recorded by the Hall effect sensor.

After finding the peaks in the data gathered from the Hall effect sensor it is possible to calculate the velocity. By comparing it with the measured values it can be seen that some minor discrepancies can be found, as shown in Fig.[6b].

The discrepancies in the data can be addressed to the fact that the measured velocity has more data points compared to the one calculated through the Hall effect sensor.

In Fig.[6c] the behavior of the lateral acceleration of the vehicle is shown. In order to obtain the lateral acceleration, it was needed to multiply the yaw rate and the longitudinal velocity values obtained directly from the database.

The rapid and numerous changes in the behavior of the acceleration, such as sudden increase or decrease show how the car behaved in the corners of the track.

Similarly in Fig.[6d] the behavior of the longitudinal acceleration is shown.

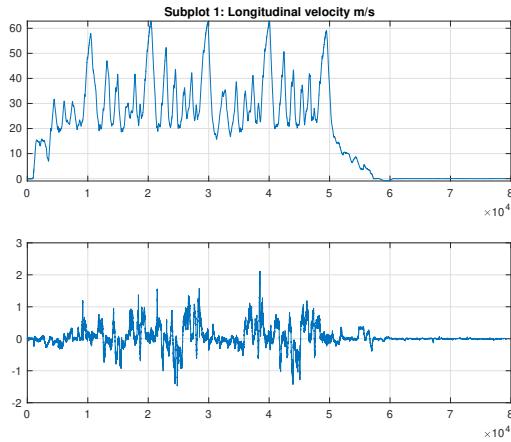
It is worth noting how the results strongly resembles the ones obtained in Fig[6b], this is due to the fact that the same raw data was used for calculating the longitudinal acceleration of the vehicle.

Finally in Fig[6e] the side slip angle behavior is evaluated and plotted.

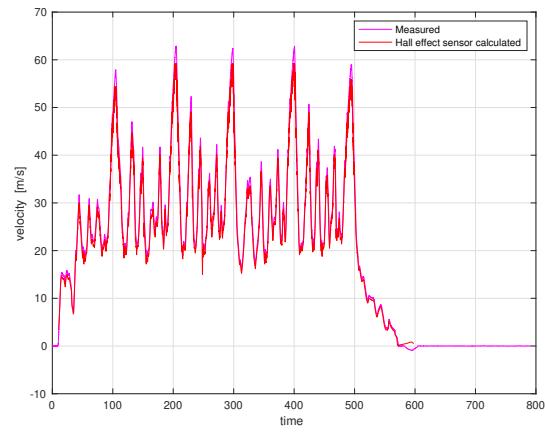
It can be observed through the graph that the side slip is generally very low in most circumstances, this highlights how the vehicle does not over-steer much, therefore it could even be assumed that the vehicle is either front-wheel driven or AWD.

A small or nonexistent oversteer is crucial for sports car as they require to take corners as fast as possible

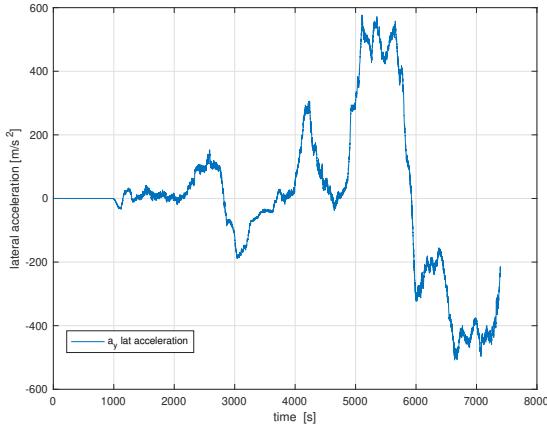
and doing that with a strong oversteer tendency might result in loosing traction and loosing control of the vehicle, thus a neutral steering behavior would be the best configuration.



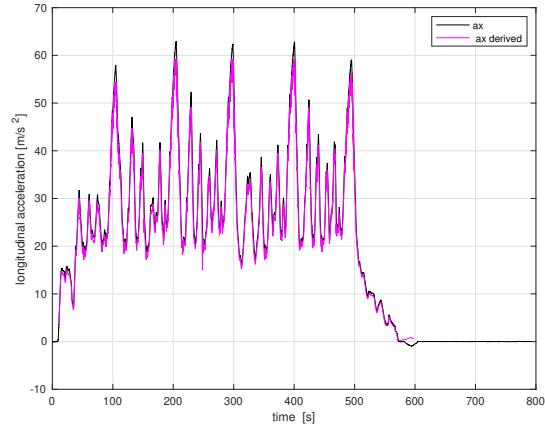
(a) Longitudinal on top and lateral velocities on the bottom



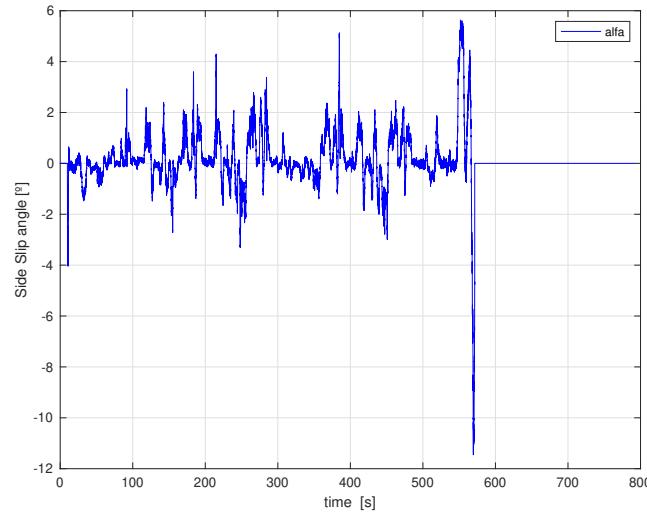
(b) measured vs Hall effect data based velocity plot



(c) Lateral acceleration calculated with the yaw rate and the longitudinal velocity



(d) Longitudinal acceleration measured vs Hall effect sensor



(e) side slip angle

Figure 6: vehicle data parameters behaviors

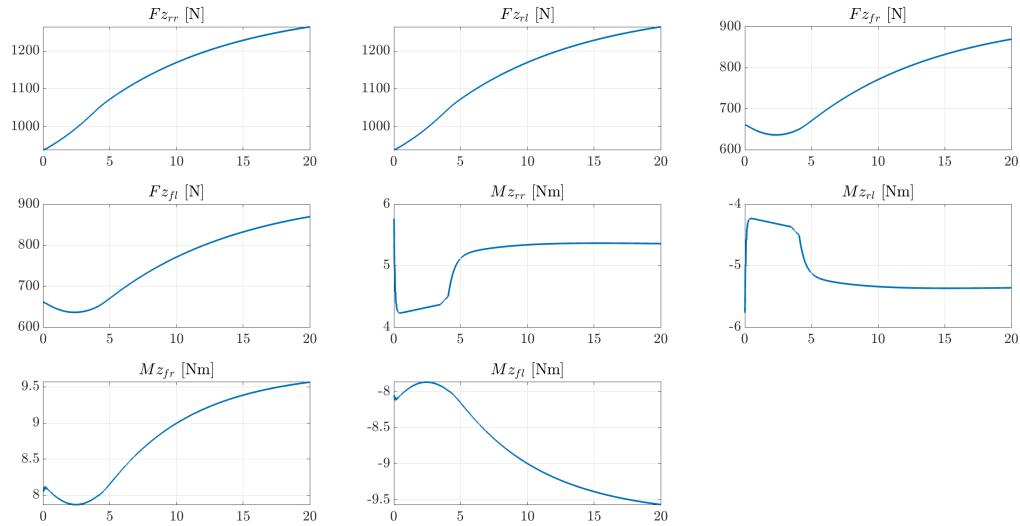
4 Exercise 4

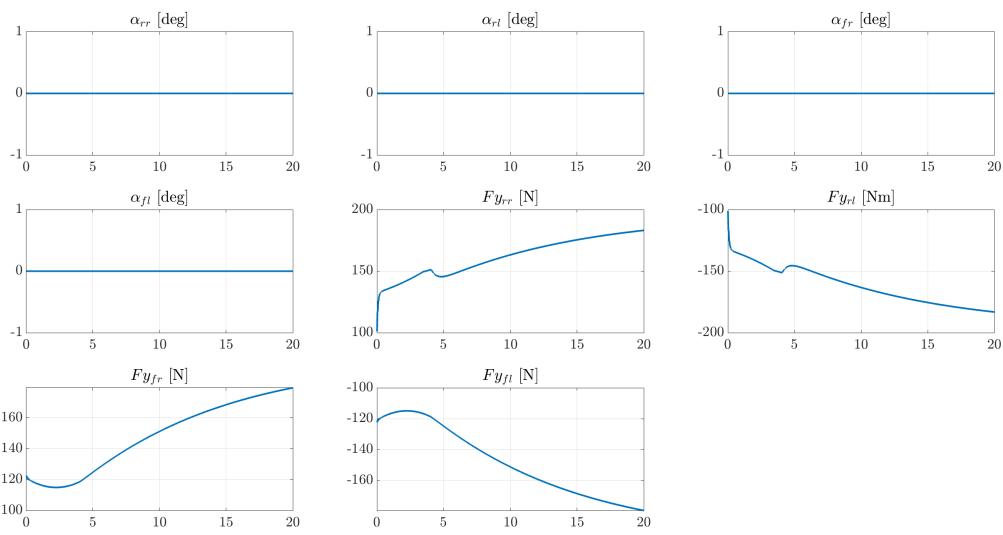
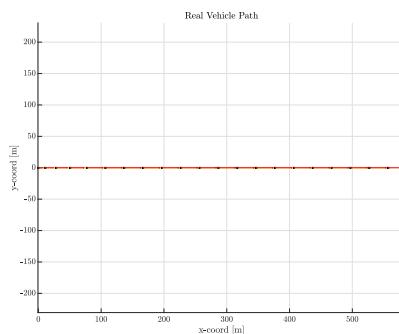
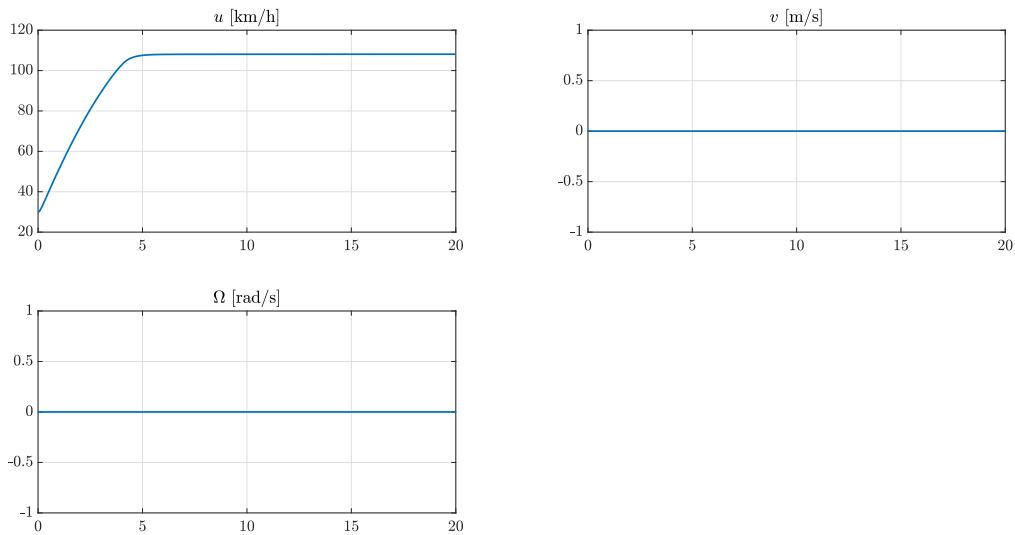
After implementing the missing equations following the slides provided during the lectures the system was tested under 3 different conditions. One where the car goes on a straight line at full throttle, the second one where the car given a high initial velocity quickly and strongly brakes at full power while maintaining a straight trajectory and the final scenario where the car constantly turns with an angle of 20 deg.

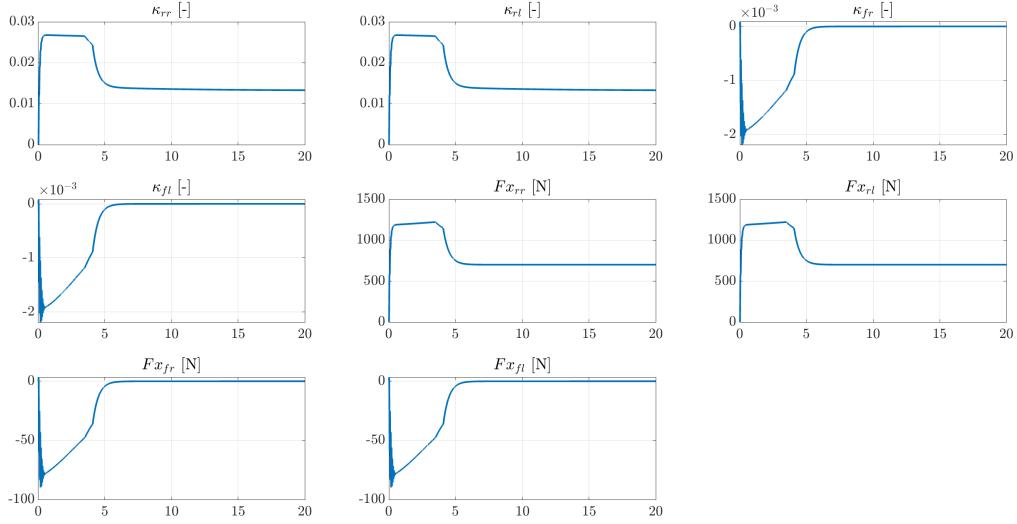
4.1 Scenario 1

The figures below were obtained with the given initial conditions:

1. initial longitudinal velocity equal to 30 km/h
2. a total simulation time of 20 s with a time step of 0.001 s
3. full throttle
4. 0 deg steering







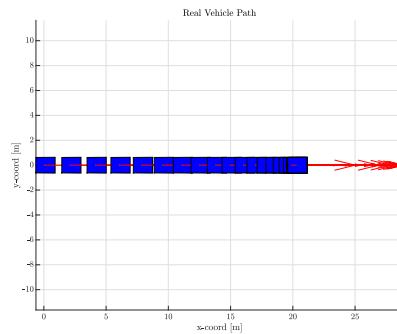
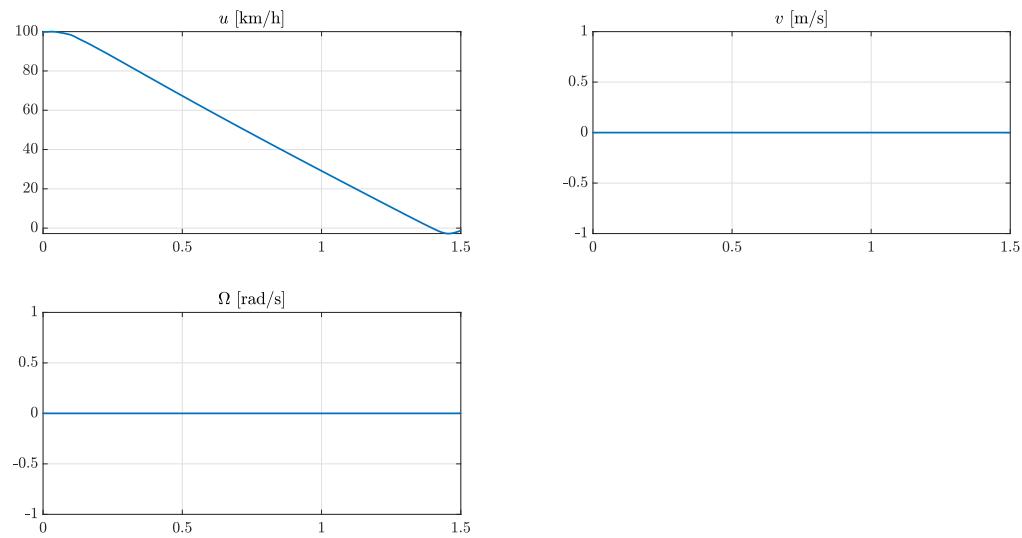
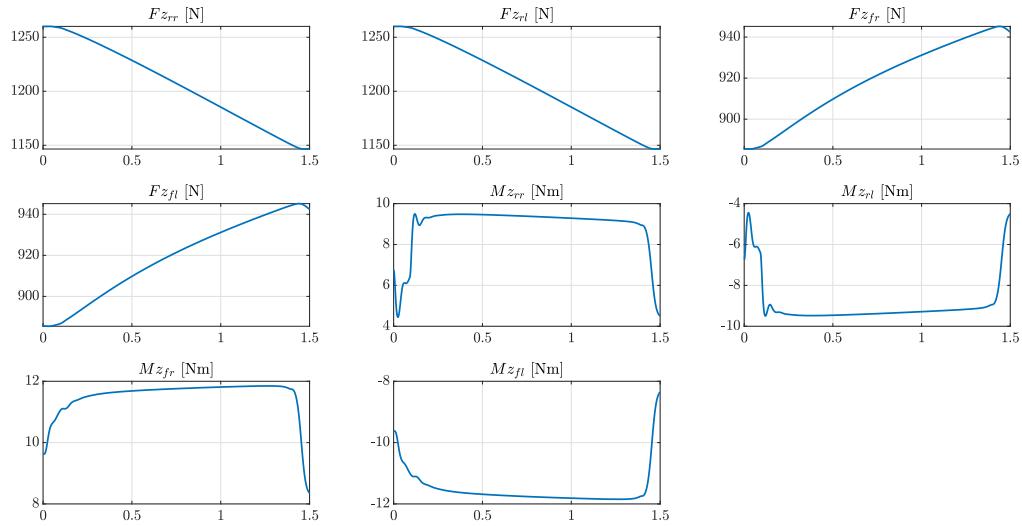
As it can be seen in the figures above, the side slip angle α is as expected 0 throughout the simulation as the vehicle follows a straight line. Conversely for what regards the longitudinal slip κ there is a sudden decrease of the value until 5s and afterwards the behavior stabilizes. This can be explained also thank to the plot representing the longitudinal velocity u , as it is shown the vehicle has a rapid increase of velocity until it reaches a constant final velocity of around 110 km/h and the stabilization of the curve of the velocity happens exactly around 5 s. A similar behavior can be observed in the behavior of the longitudinal forces F_x , after a sudden decrease given by the initialization of the model and the simulation the force rapidly increase as the car is accelerating and once the maximum velocity it is reached it stabilizes. For what regards the vertical forces F_z and moments M_z the behavior is different, as the car accelerates accelerates, there is a horizontal forward force on the car, and a corresponding backwards horizontal force on the ground. When considering the axles distribution for the forces it can be seen how during the acceleration (0-5s) the while the forces increase on the rear axle, on the front one they decrease. This indicates that the car is rear wheel driven. Conversely for what regards the moments M_z a mirrored opposite behavior for the right and left wheels of each axle can be observed. This is due to the fact that for stability constraints the moments on the left and right wheel need to balance and cancel each other. For what regards the lateral forces F_y a similar behavior for the axles distribution as for the the vertical forces during the time when the car reaches its maximum longitudinal speed can be observed. However as the car is not turning a similar mirrored opposite behavior as for the moments M_z can be observed.

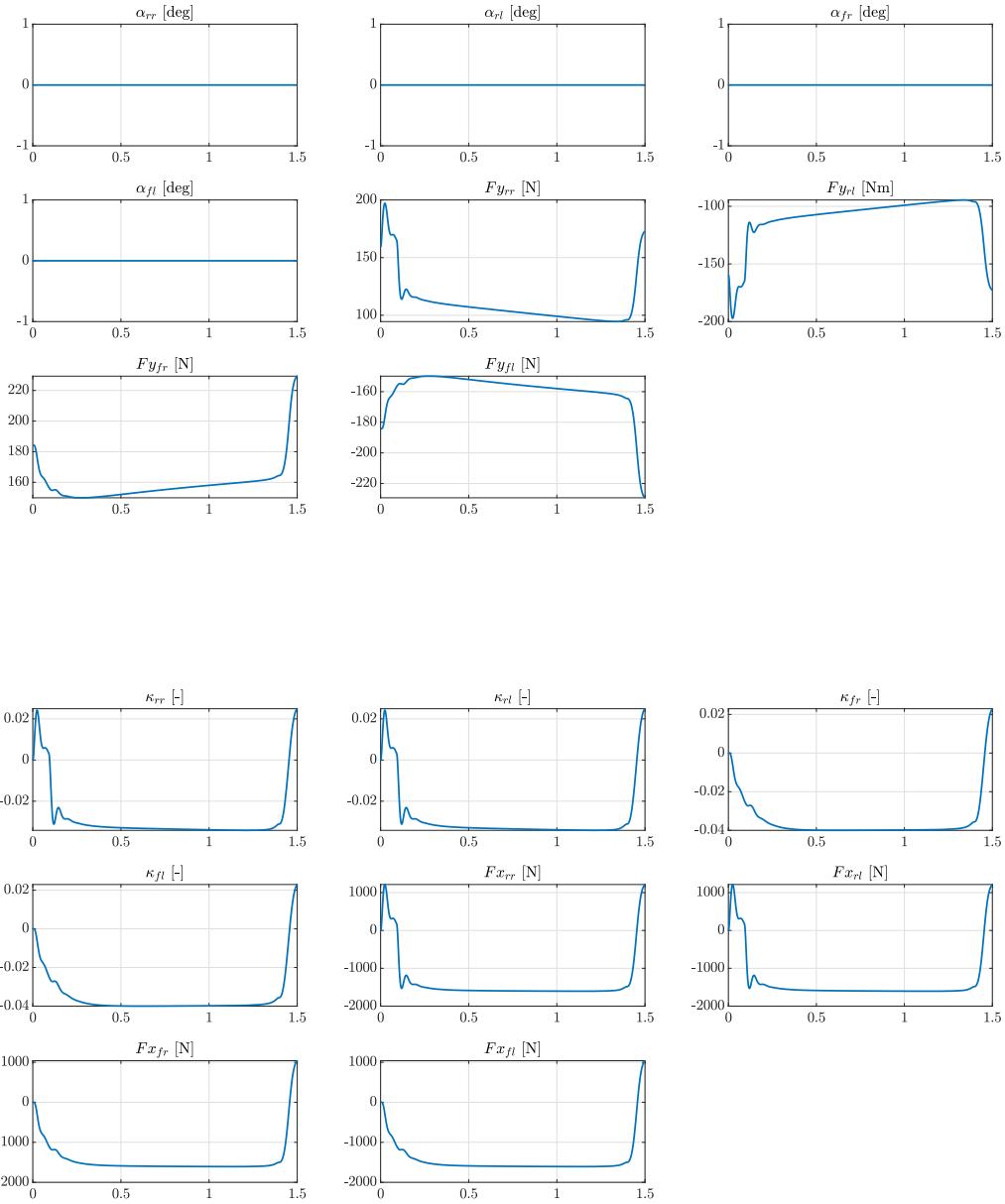
4.2 Scenario 2

In the second scenario the opposite conditions compared to the first test were to be evaluated. Given an high initial velocity the vehicle strongly brakes for a short period of time. The exact conditions are the following:

1. initial longitudinal velocity equal to 100 km/h
2. a total simulation time of 1.5 s with a time step of 0.001 s
3. full brake
4. 0 deg steering

The results obtained are showed in the figures below.





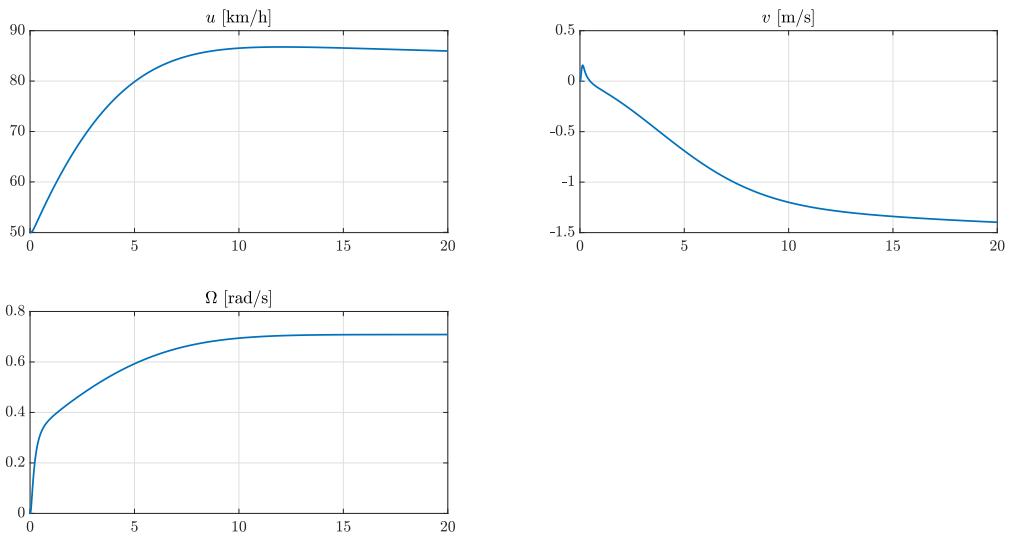
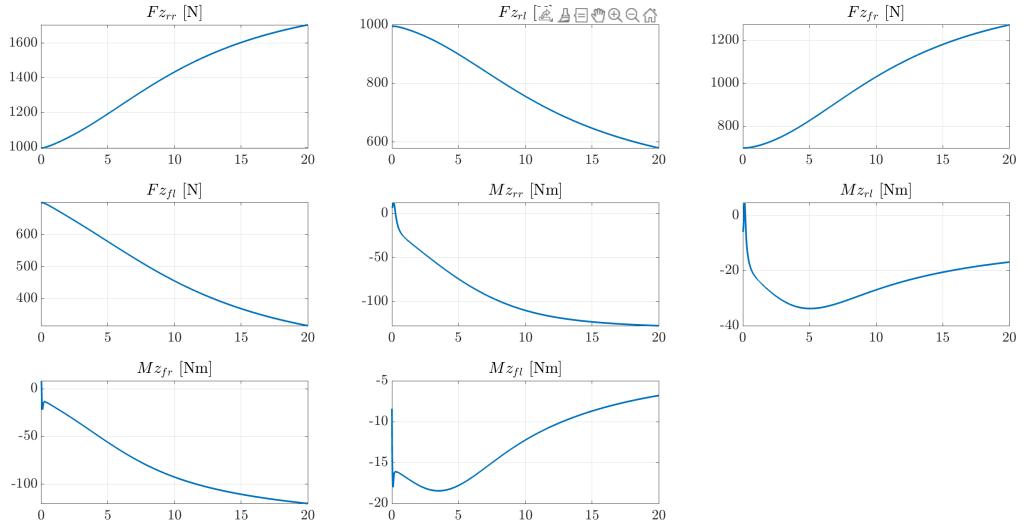
As it can be seen in the plot representing the longitudinal velocity behavior of the vehicle, the speed drastically and quickly shrinks until the vehicle is brought to stand still at the end of the 1.5s simulation time. The same pattern can be observed as well in the graph representing the vehicle's path. For what regards the vertical forces F_z it can be observed how the behavior of the axles is the opposite compared to the one observed in the previous scenario. For what regards the moments M_z instead they present a different behavior compared to the previous scenario. After an initial sudden increase or decrease, given by the starting of the simulation they maintain a linear behavior throughout the time the vehicle is decreasing the speed, while it can be noted how towards the end of the simulation there is a sudden increase for the left wheels for the and decrease for the right wheels, which highlights the stand stillness of the vehicle. For the lateral forces F_y a very similar behavior is followed. Also in this case as the vehicle is going on a straight line α is constantly 0. For what regards the longitudinal slip κ it can be seen how it is negative throughout the entire time the

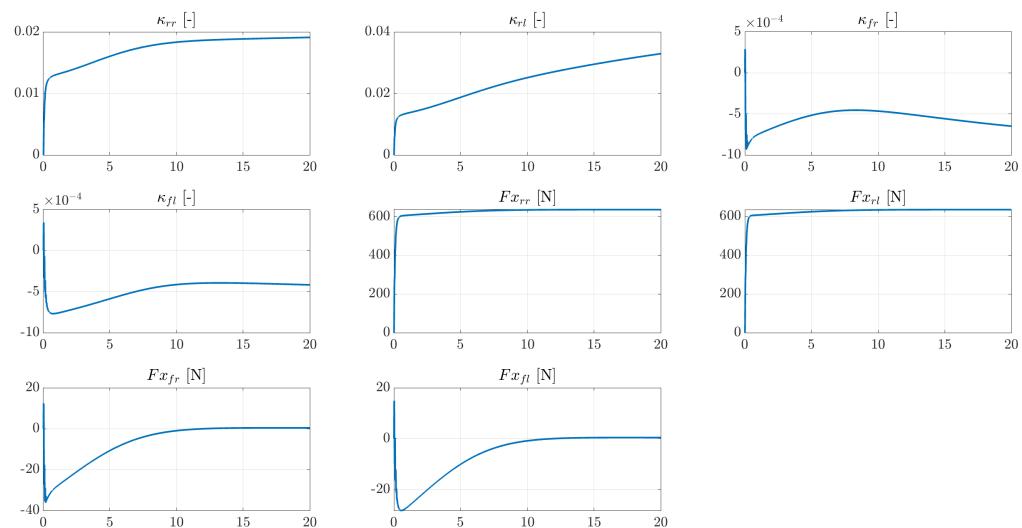
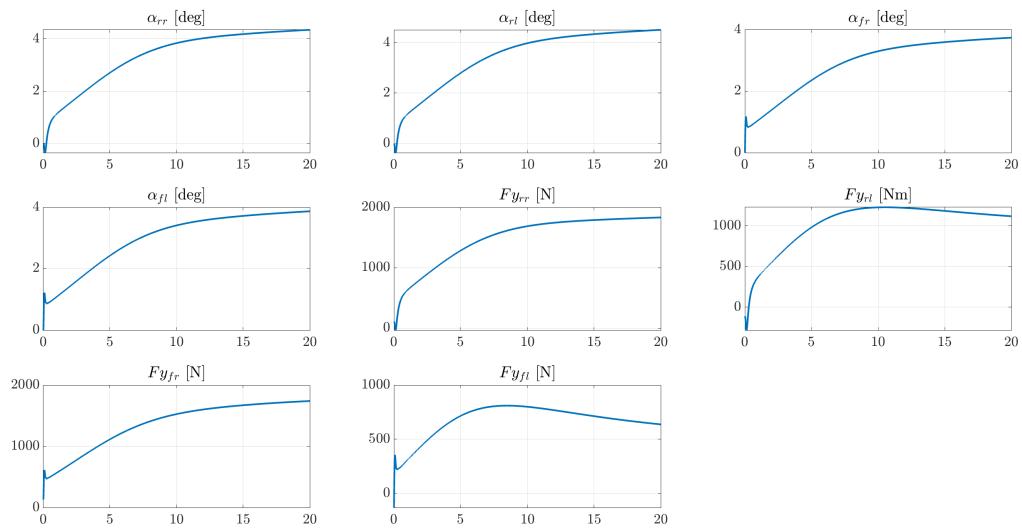
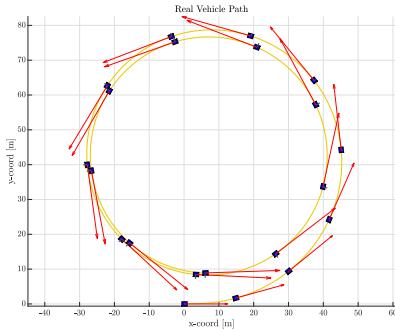
vehicle is braking effectively. Similarly as we are braking thus there is a negative longitudinal acceleration, the longitudinal forces F_x are negative as well.

4.3 Scenario 3

In this last scenario the vehicle is tested under 50% of the available throttle and constant steering behavior. The exact given conditions are the following:

1. initial longitudinal velocity equal to 50 km/h
2. a total simulation time of 20 s with a time step of 0.001 s
3. 50% throttle
4. 20 deg steering





As the vehicle is accelerating the same behavior for the vertical forces F_z as the one observed in the first scenario is shown in the figures above. Similarly the longitudinal velocity increases for the first 10 s of the simulation until it stabilizes. A similar behavior as observed in the first scenario can be seen also for the longitudinal forces F_x and longitudinal side slip κ as the vehicle is accelerating. Conversely a complete different behavior is observed for the side slip angle α and the lateral forces F_y as the vehicle is constantly steering the forces increase throughout the entire simulation time as well as the side slip angle however there is a stark difference between the right and left wheels. The right wheels experience a maximum force of 2000 N while the left wheels experience a maximum of 1000 N, this is due to the weight transfer from 'inside' to the 'outside' whilst cornering.

5 Assignment 5

5.1 Exercise 5.1

In order to obtain the vehicle handling diagram shown in the figures below the Matlab function *polyfit* was employed. Several tests with increasing number of polynomials were done, however the diagram could be satisfactorily fit with a first degree polynomial. The obtained images show the results of a first degree polynomial fitting. The vehicle handling diagram as it can be seen in the graphs displayed in Fig.[8] and Fig.[9] have been obtained by plotting the lateral acceleration a_y on the x-axis and the equation below on the y-axis:

$$\delta - \frac{\omega \cdot L}{u}$$

where δ represents the steering angle, u represents the longitudinal velocity, ω the yaw rate of the vehicle and L the vehicle wheelbase.

5.1.1 Exercise 5.1.1

For this first test the steering angle δ was kept low at only 5 deg in order to perform sine steer maneuvers. The obtained graphs below have been produced using a total simulation time of 40 s, a frequency of 0.1 Hz and a time step of 0.1. As it is shown in Fig[8] the vehicle behavior is clearly oversteering, this can be seen by how the value of the understeer coefficient K_{us} being negative affects the fitted curve. It is worth noting how with an increase of the velocity of the vehicle the handling diagram gradually becomes more steep, this highlights how at higher velocities the oversteering becomes more severe.

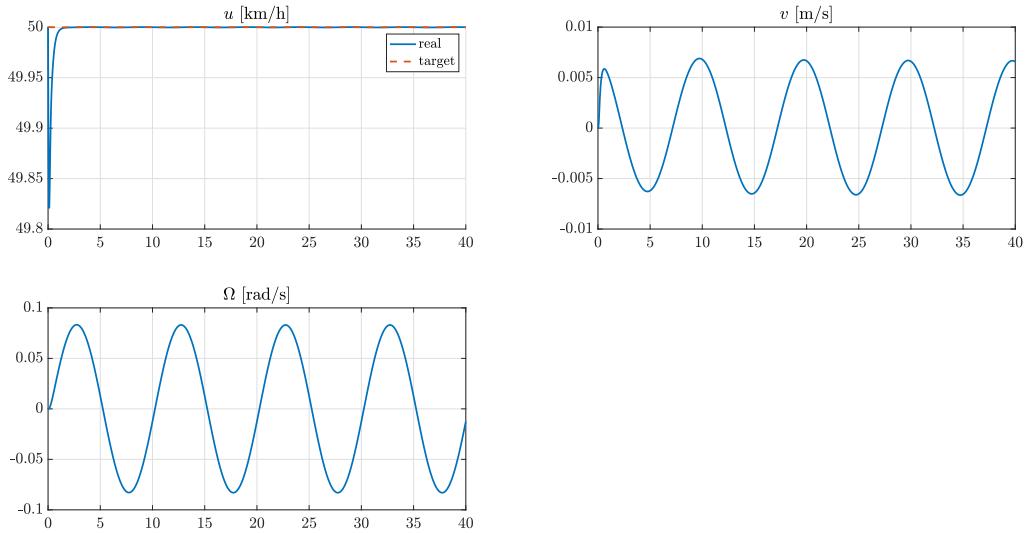
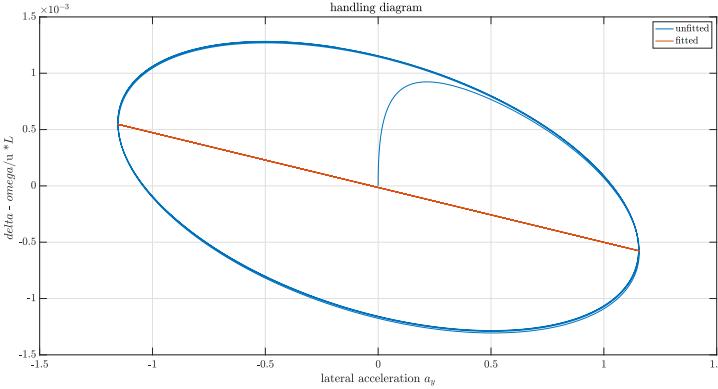
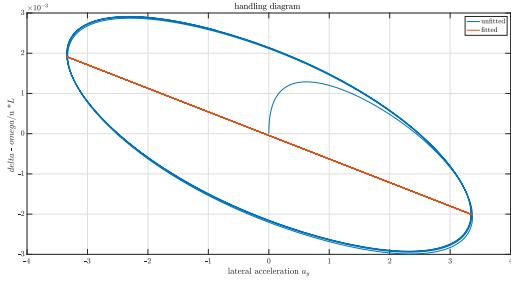


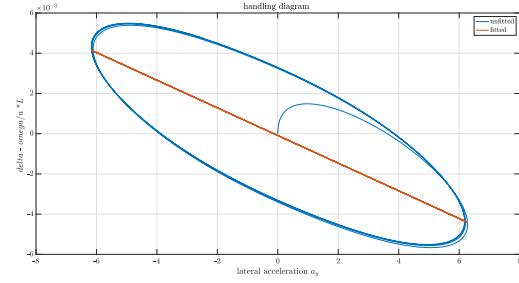
Figure 7: Vehicle longitudinal and lateral velocities and yaw rate at 50 km/h



(a) 50 km/h



(b) 80 km/h

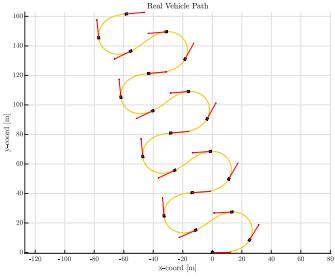


(c) 100 km/h

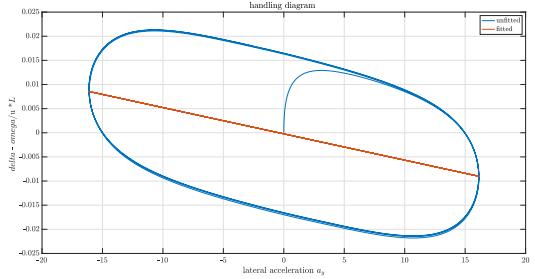
Figure 8: Illustration of various handling diagrams for different speeds and same sine steer maneuver

5.1.2 Exercise 5.1.2

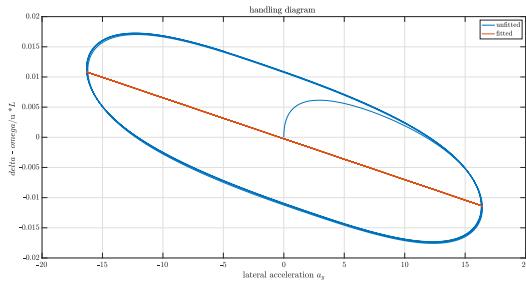
The new tests do not substantially bring new information regarding the vehicle handling as it shows also in these tests a clear oversteering behavior. The fitting was done using the same *polyfit* function which was employed in the previous task with a first degree polynomial. The three coefficient K_{us} do not have the same value as the steepness of the curve is speed and angle dependant and the three test were all performed with different conditions.



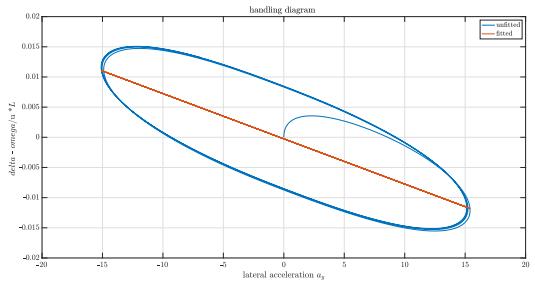
(a) vehicle path at 50 km/h



(b) $\delta = 70\text{deg}$, $u = 50\text{km/h}$



(c) $\delta = 24 \text{ deg}$, $u = 80 \text{ km/h}$



(d) $\delta = 12 \text{ deg}$, $u = 100 \text{ km/h}$

Figure 9: vehicle handling with different sine steer maneuver at different speeds and steering angles

5.2 Exercise 5.2

As it can be seen in figures below depicting the steering behavior of the car and most importantly of the vehicle path, it is possible to understand the general vehicle behavior at constant steer maneuvers. Similarly to the previous exercise the slope of the vehicle handling indicates an oversteering tendency of the vehicle. Moreover this is accentuated in Fig.[12d]where the turning performed by the vehicle show a remarkable concentric behavior. The constant steer maneuver is undoubtly more feasible to achieve in a real life setting to understand the general behavior of the car, despite the fact that in a simulated environment a sine maneuver with a small steering angle is able to create a more 'straight' like path, therefore it allows to accurately understand if the vehicle is oversteering or understeering.

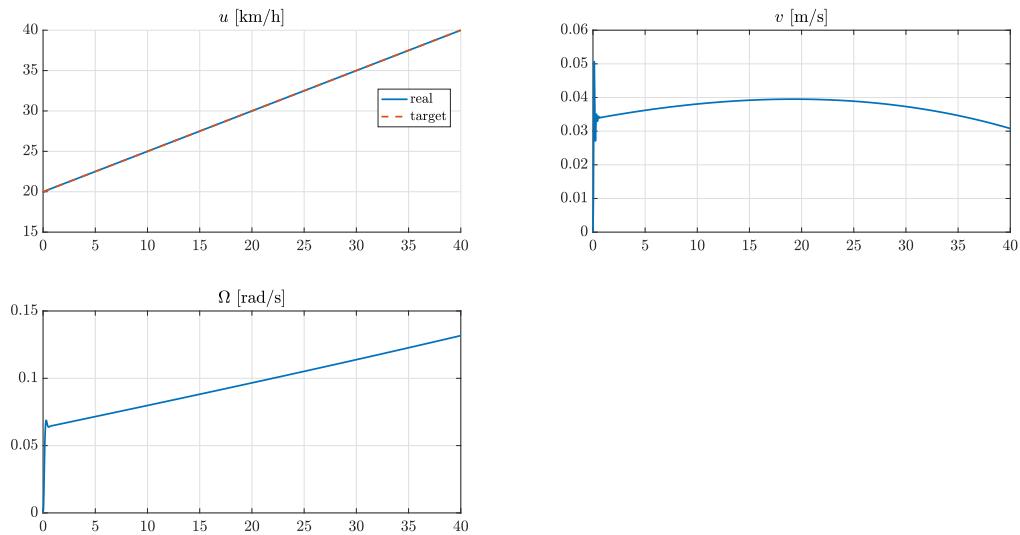


Figure 10: Constant steer speed profile $u_i = 20$ km/h, $u_f = 40$ km/h

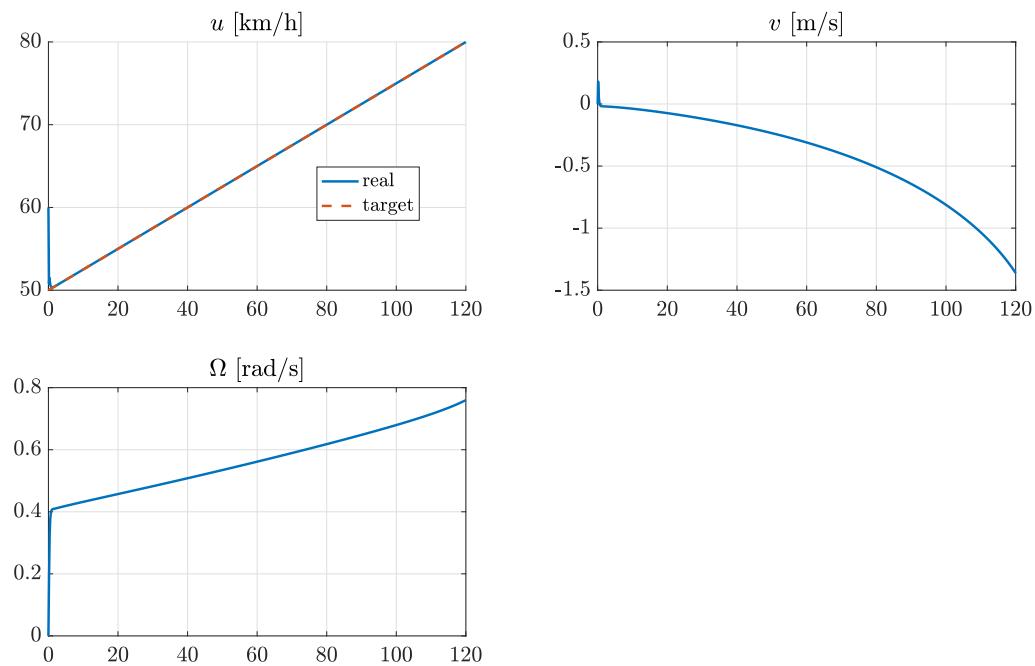


Figure 11: Constant steer speed profile $u_i = 50$ km/h, $u_f = 80$ km/h

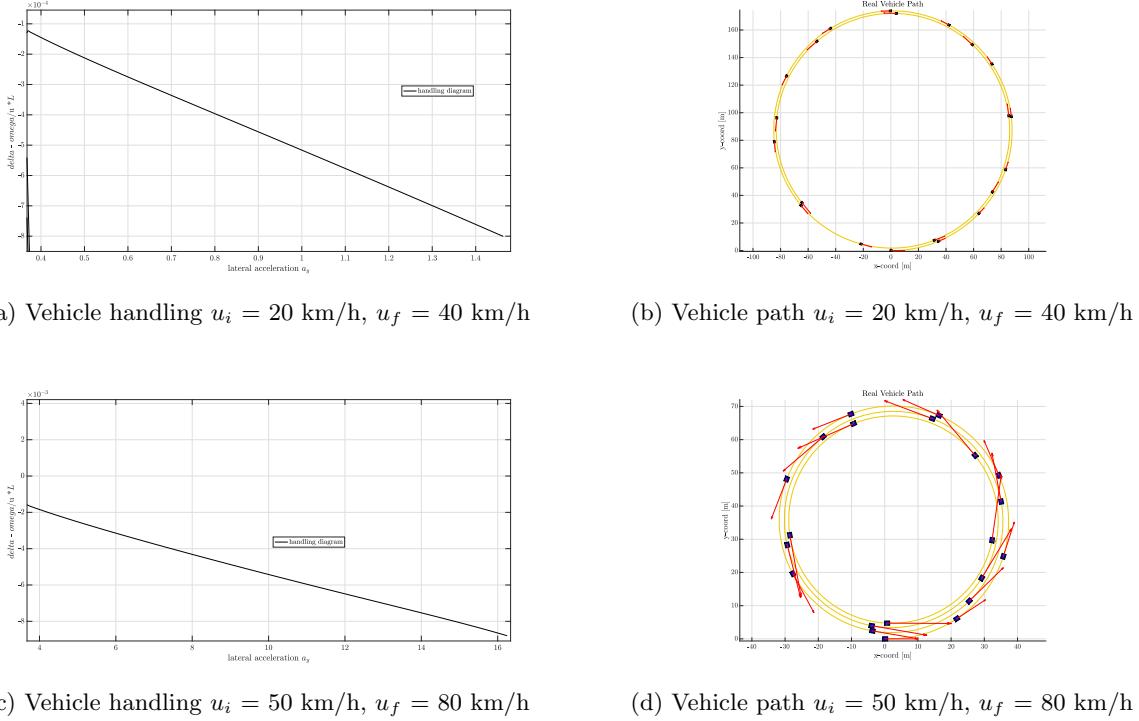


Figure 12: Illustration of vehicle handling and path at constant steer maneuvers

6 Assignment 6

In this assignment the task was to evaluate the performance of 4 different lateral controllers. In order for the tests and the comparison among the controllers to be the more reliable as possible it was decided to perform for each of them 3 different maneuvers at 3 different speeds, namely 30, 50 and 60 km/h and whenever the controller was able to in a way to follow the path at higher speeds also at 80 km/h. However this last test was performed successfully only by the Stanley Dynamic controller

6.1 Clothoid Controller

The first controller to be tested was the Clothoid based controller. As the tuning parameters for the controller are mainly the lookahead distance and the understeer gradient K_{us} the first required step was evaluating the K_{us} coefficient at different speed on the basis of the findings of the previous assignment. In the table below the obtained values for K_{us} are presented.

Table 1: Understeer gradient K_{us}

Speed [km/h]	K_{us}
20	-5.062935410300344e-04
30	-4.710879256496287e-04
40	-4.715285981810922e-04
50	-4.865648970989296e-04
60	-5.114163963830335e-04
80	-5.855245738485693e-04
100	-6.875897089988750e-04

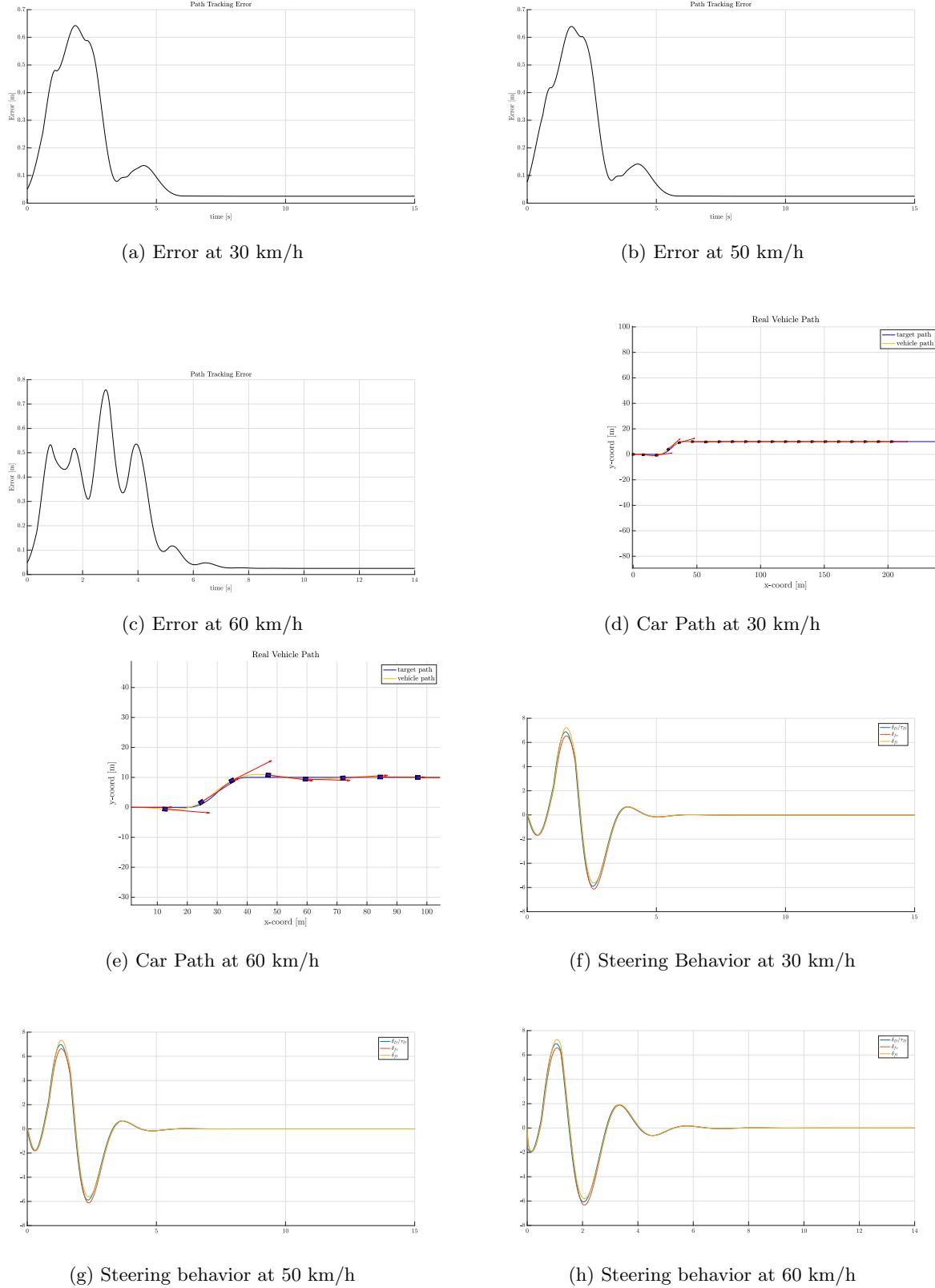


Figure 13: Clothoid based controller tests data

In Fig.[13] the obtained results are presented. As it can be seen in Fig.[13a] and Fig.[13b] the vehicle is able to follow the given path at both 30 and 50 km/h with relative ease and a low max and mean error, as it is presented in Tab.[2]. The same can be said also for the steering behavior as the settling time for both results at 30 and 50 km/h is reached without heavy oscillations. On the contrary an higher velocity results not only an higher path tracking error, where the most substantial difference is observed for the mean error as it is also shown by the behavior of the controller in both Fig.[19c] and Fig[19e]. Moreover the increased number of oscillations in Fig.[13h] before the system reaches the steady state error underline how the controller with an higher velocity has a more underdamped behavior. Overall the Clothoid controller showed very good results compared to the other 3 controllers.

The main drawback which emerged through the tests is that at high speeds regardless of the lookahead distance it is not possible to stabilize the controller behavior after the sharp turn, either it cuts it (high lookahead) or it behaves like a spiral after it or a wave like behavior (lower values = 20-25 range).

Table 2: Clothoid Controller Error Data

Speed [km/h]	lookahead [m]	Max Error [m]	Mean Error [m]
30	20	0.64245	0.12091
50	20	0.63631	0.11703
60	23	0.75824	0.16505

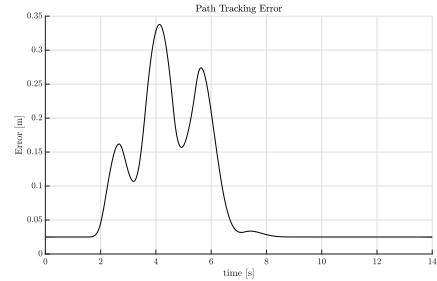
6.2 Arc Path Controller

The second controller tested was the Arc Path controller. The equations presented during class were used for its implementation below. It is worth noting that there was the need to implement a rotational matrix in order to align the points counterclockwise in order for the controller to produce a path on the same framework with respect to the designated path.

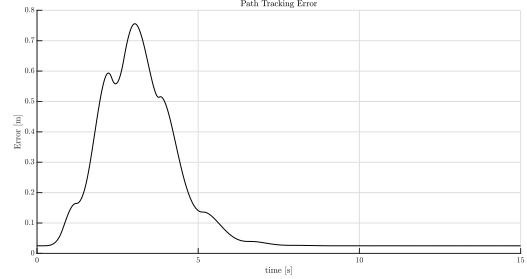
```

1 function delta = arcPathFollowController(target_Point, pose, endOfCircuit)
2
3 % Load vehicle params
4 vehicle_params = getVehicleDataStruct();
5 Lf = vehicle_params.vehicle.Lf;
6 Lr = vehicle_params.vehicle.Lr;
7 L = Lf + Lr; % [m] vehicle wheelbase
8
9 % Current vehicle center of mass pose
10 x_C = pose(1);
11 y_C = pose(2);
12 theta_C = pose(3); % vehicle attitude (in radians)
13
14 %% COMPLETE THIS WITH YOUR CODE
15
16 if (~endOfCircuit)
17 rmat = [cos(theta_C) -sin(theta_C); sin(theta_C) cos(theta_C)] ;
18
19 P = ([target_Point(1) target_Point(2)] -[x_C y_C]) *rmat;
20 xd = P(1)+Lr;
21 yd = P(2);
22 lam = atan(yd/xd);
23 R = sqrt(xd^2+yd^2)/(2*sin(lam));
24
25 delta = atan(L/R);
26 else
27 delta = 0;
28 end
29
30 end

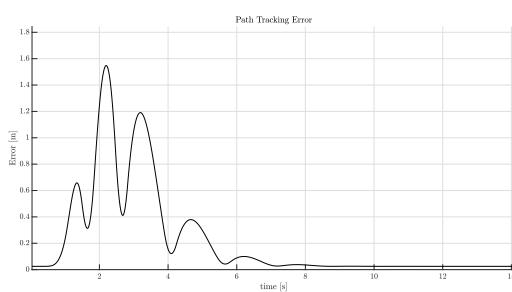
```



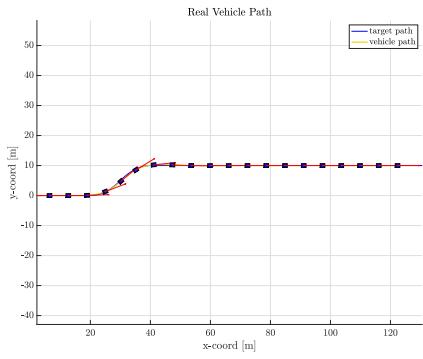
(a) Error at 30 km/h



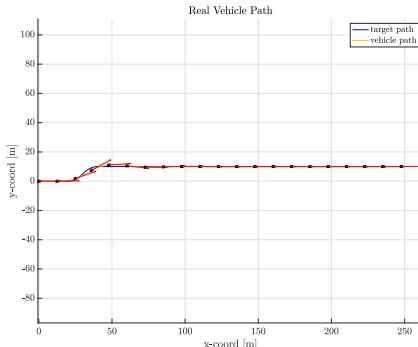
(b) Error at 50 km/h



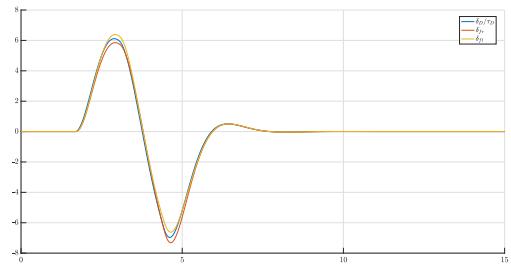
(c) Error at 60 km/h



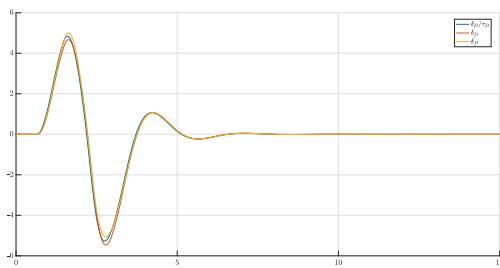
(d) Car Path at 30 km/h



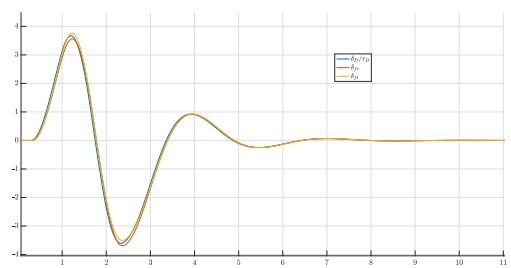
(e) Car Path at 60 km/h



(f) Steering Behavior at 30 km/h



(g) Steering behavior at 50 km/h



(h) Steering behavior at 60 km/h

Figure 14: Arc Path based controller tests data

Table 3: Arc path Controller Error Data

Speed [km/h]	lookahead [m]	Max Error [m]	Mean Error [m]
30	6	0.33781	0.08024
50	11	0.75580	0.14647
60	15	1.54807	0.32442

In Fig.[14] and in Tab.[3] the obtained results are presented. Similarly to the Clothoid controller for the Arc Path controller there is the need to tune the lookahead distance. As it can be seen through the results obtained the Arc Path controller performs better compared to the Clothoid one at lower speed, in our case at 30 km/h. At 50 km/h the two controllers have similar results, while at 60 km/h as it is also shown in Fig.[14h]. Overall by performing other tests as well on the Arc Path controller the general observed behavior is that if the lookahead distance is increased it tends to cut the curve, if you decrease too much the steering behavior is negatively effected. It is less linear than what showed in Fig.[14d] and Fig.[14e], thus more sharper corrections are required in order to follow the path and the settling time is therefore negatively affected as well.

6.3 Stanley Kinematic

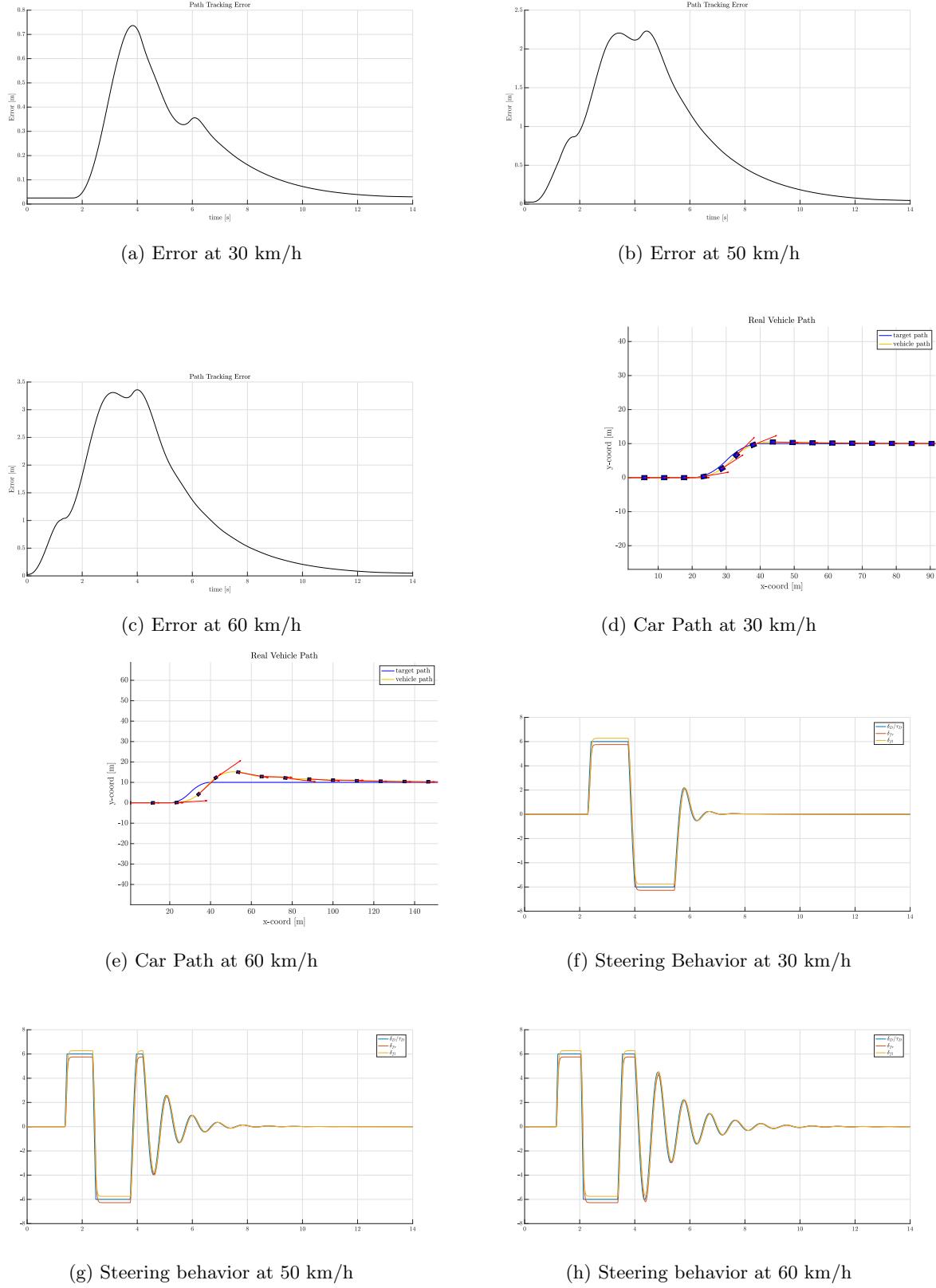


Figure 15: Stanley Kinematic controller tests data

Table 4: Stanley Kinematic Controller Error Data

Speed [km/h]	Max Error [m]	Mean Error [m]
30	0.7366	0.19563
50	2.23041	0.75582
60	3.35938	1.07395

The third analyzed controller was the Stanley Kinematic controller. Conversely from the previous two controllers the parameters to tune are completely different. One is the maximum steering angle at the front wheel. For control purposes it is kept low and with a value minor compared to the real maximum steering angle. For these tests the angle was kept at 6 deg as it showed the best overall results. The second parameter used in the tuning of the controller is the position gain for the forward motion.

As it is shown in Fig.[16] and in Tab.[4] the Stanley Kinematic controller does not present any substantial advantage compared to the previous two controllers. The steering behavior with an increase of the speed presents a severely underdamped and higher settling time as shown in Fig.[15h] for instance.

6.4 Stanley Dynamic

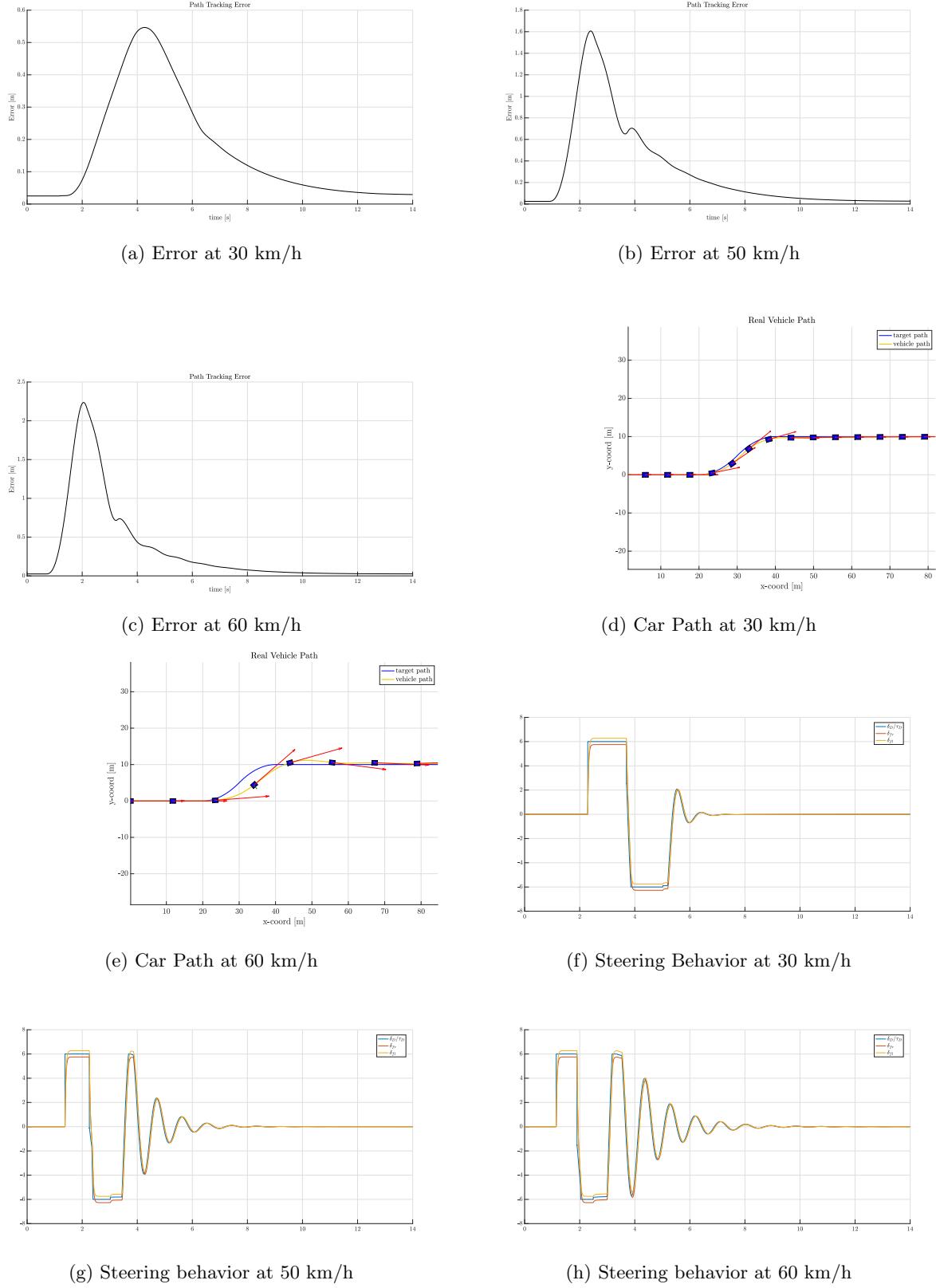
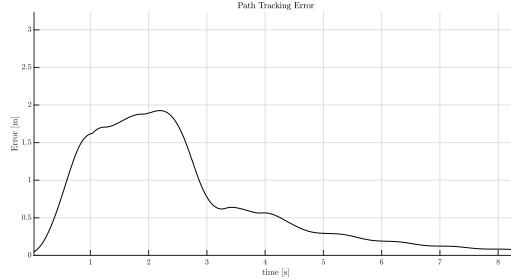
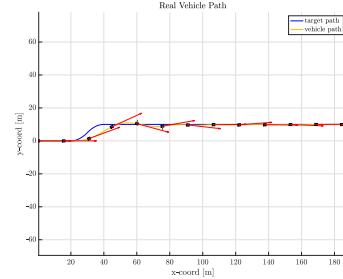


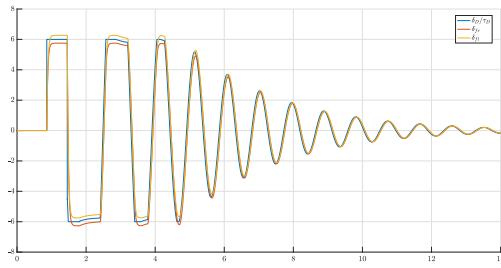
Figure 16: Stanley Dynamic controller tests data



(a) Error at 80 km/h



(b) Car Path at 80 km/h



(c) Steering Behavior at 80 km/h

Figure 17: Stanley Dynamic controller data at 80 km/h

Table 5: Stanley Dynamic Controller Error Data

Speed [km/h]	Max Error [m]	Mean Error [m]
30	0.54645	0.16273
50	1.60675	0.30433
60	2.23615	0.32881

The last analyzed controller for this assignment was the Stanley Dynamic controller. The main difference compared to the Stanley Kinematic is that it is tuned through the yaw rate feedback gain and the steering angle feedback gain. Overall the obtained results resembles the ones obtained with the Stanley Kinematic controller. However it is worth noting that this was the only controller which was able to perform the turn at high speed, 80 km/h, being able to stabilize the vehicle in the straight path following it. The results obtained are shown in Fig.[17], despite the steering behavior being not linear and underdamped with a high settling time, the vehicle as shown in Fig.[17a] and Fig.[17b] was able to follow the path despite an higher path tracking error and not being able to follow the trajectory accurately as at lower speeds.

7 Assignment 7

7.1 Exercise 7.1

For this task the main goal was the optimization of the RRT* algorithm. It is a widely popular and efficient sampling method used for roadmap creation, however as every sampling algorithm the computational time is negatively affected in case the number of samples are increased. Therefore a trade-off needs to be found in order to create a good roadmap with low computational times. The optimization approach followed was based on a trial and error approach, with its main goal the decrease of the min and max number of iterations. At the same time the max steering angle was intentionally kept low at 4 deg. The final path planner function was optimized with the following values:

- `planner = pathPlannerRRT(costmap,'ConnectionDistance',40, 'GoalBias',0.1, 'MinIterations',1e3,'MaxIterations',1e4, 'MinTurningRadius',min_turn_radius);`

Additionally the interpolation of the calculated reference path was kept low with a value equal to 10% of the sampling distance. In this way the obtained path was computed by the RRT* in 0.6 s and the total simulation time was always in the range around 15-25 s. The path showed in the next exercise were obtained using a Clothoid fitting of the path.

It was decided to use a clothoid fitting instead of the Dubins curves for several reasons. Most of all the Dubins curves are composed by 3 arcs, either a left or right curve or a straight line, however the curves are always of the maximum curvature, therefore with an increased speed it might become challenging in a real life setting to maintain control of the vehicle. In this way when the speed is increased the vehicle is able to follow the path better thanks to smoother trajectories.

The smoother the path is the better the vehicle is able to follow it and it is also able to complete the path in a lower amount of time. For this reason it is important to use the clothoids fitting as it greatly helps to balance the quality of the path.

7.2 Exercise 7.2

In the following exercise the two controllers which showed the best results in the previous assignment were evaluated. As the paths presented sharp turns it was decided to evaluate the paths only at low and medium speed, namely at 20 km/h and 40 km/h.

7.2.1 Arc-Path Controller

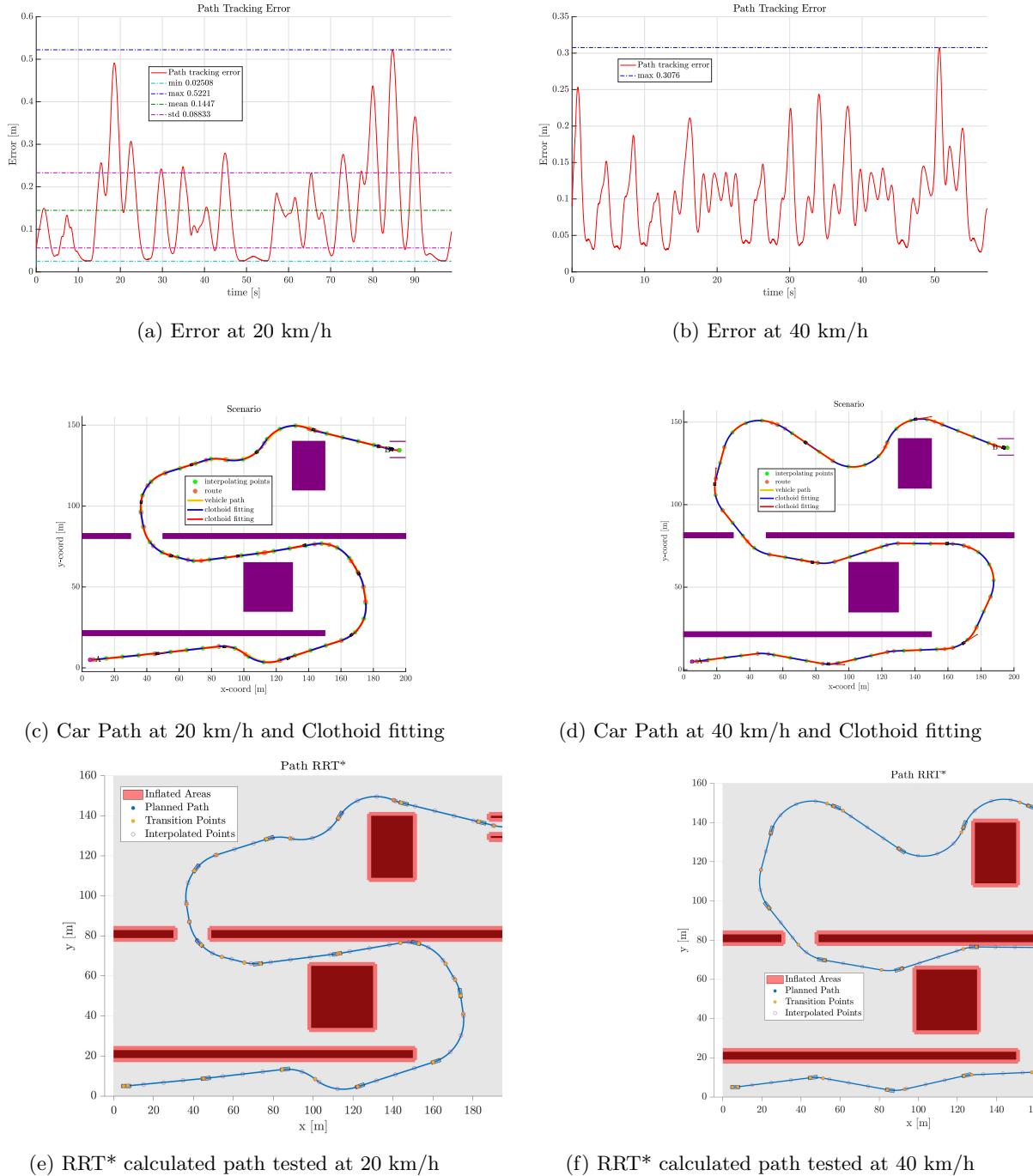


Figure 18: Arc Path controller tests data

The obtained graphs above were computed and evaluated in the simulation at 20 km/h with the following data:

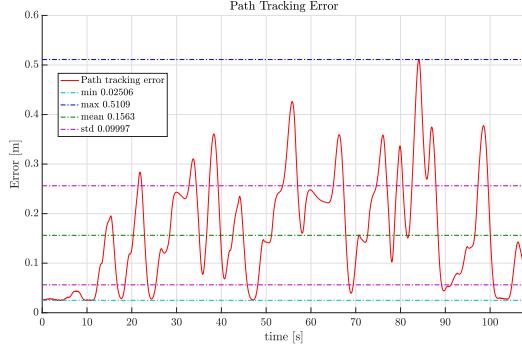
- lookahead distance: 10 m
- RRT* computation time: 0.4 s
- total simulation time: 27.5 s
- time taken by the vehicle to complete the scenario: 98.748 s

For the test at 40 km/h, the following data was tuned and given as output by the simulation.

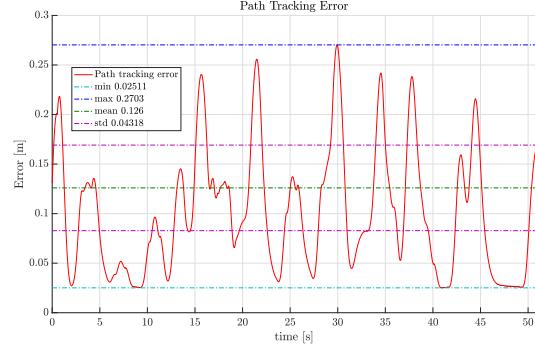
- lookahead distance: 8 m
- RRT* computation time: 0.2 s
- total simulation time: 21.7 s
- time taken by the vehicle to complete the scenario: 49.543 s

The most notable comparison regards the maximum error, as it can be seen the maximum value of the path tracking error at 40 km/h is lower compared to the value obtained at 20 km/h. However this can be explained by the stark difference in the path smoothness as it is shown in Fig.[18]. However when comparing the general behavior of the path tracking error it can be seen how at higher velocity there is an higher incidence of local maxima values, while at 20 km/h a more smooth and linear behavior can be observed. Therefore at higher speed the path tracking error behavior indicates how the controller steering behavior had to perform sharper corrections compared to the low speed test.

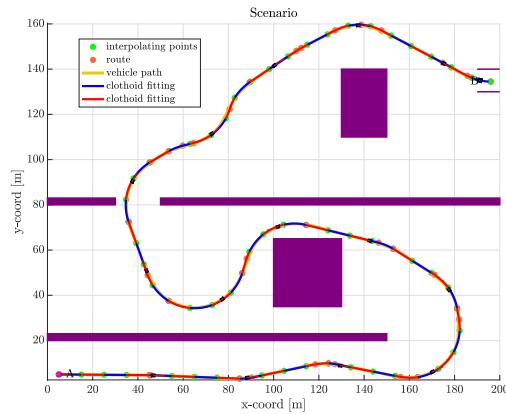
7.2.2 Clothoid controller



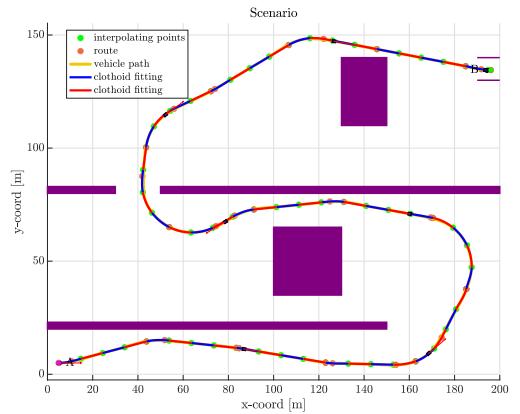
(a) Error at 20 km/h



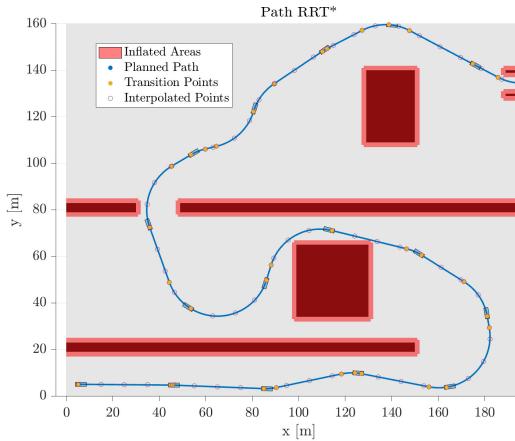
(b) Error at 40 km/h



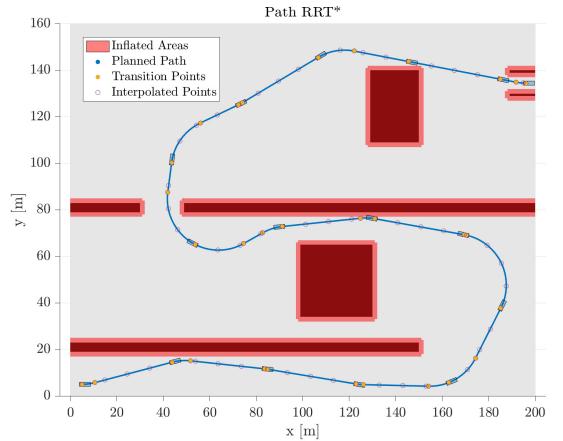
(c) Car Path at 20 km/h and Clothoid interpolation



(d) Car Path at 40 km/h and Clothoid interpolation



(e) RRT* calculated path tested at 20 km/h



(f) RRT* calculated path tested at 40 km/h

Figure 19: Clothoid controller tests data

Two tests were also performed for the clothoid based controller. Namely at 20 and 40 km/h.

- speed: 20 km/h
- lookahead: 14 m
- RRT* computation time: 0.5 s
- total simulation time: 64.1 s
- time taken by the vehicle to complete the scenario: 107.976 s

As it can be seen through the presented data the total computing time for the simulation was considerably higher compared to the results of the pure pursuit controller. Compared to the Arc Path controller the needed lookahead distance to obtain a low maximum path tracking error is higher. When analyzing the general path tracking error behavior it can be seen how also in this case at higher speed the graph presents a high number of local maxima, thus meaning that at higher speeds the controller had to perform sharper correctional turns to follow the path.

- speed: 40 km/h
- lookahead: 14 m
- RRT* computation time: 0.4 s
- total simulation time: 87.7 s
- time taken by the vehicle to complete the scenario: 50.814 s

A Assignment 2 Matlab code

A.1 Exercise 2.1

```
1 %% Plot the data
2 figure(1);
3 subplot(5,1, 1);
4 plot(SL);
5 grid on
6 title('longitudinal slip')
7 subplot(5,1,2);
8 plot(SA, 'k');
9 grid on
10 title('side slip')
11 subplot(5,1,3);
12 plot(IA);
13 grid on
14 title('camber angle');
15 subplot(5,1,4);
16 plot(FZ);
17 grid on
18 title('Vertical tire force')
19 subplot(5,1,5);
20 plot(P);
21 grid on
22 title('Pressure')
23 %% Organize the data
24 % 1. create a vector of idxs dividing the dataset by parameters
25 DT = horzcat(FX, FY, FZ, MZ, SA, SL, P, IA);
26
27
28 %Ia = gamma
29 % SA = alpha
30
31 FX1 = FX; FX2 = FX; FX3 = FX;
32 FZ1 = FZ;
33 SL1 = SL; SL2 = SL; SL3 = SL;
34 RowtoD = abs(SA)> 0.04 | IA > 0.04; % range
35 RowtoD1 = IA > 0.04 | abs(SA) < 2.8 | abs(SA) > 3.1;
36 RowtoD2 = IA > 0.04 | abs(SA) < 5.8 | abs(SA) > 6.1;
37
38 FX1(RowtoD) = []; FX2(RowtoD1) = [] ;FX3(RowtoD2) = [];
39 FZ1(RowtoD) = [];
40 SL1(RowtoD) = [] ; SL2(RowtoD1) = [] ; SL3(RowtoD2) = [];
41 % deleting the rows where alpha and gamma are non zero
42
43 % 670 load % 950 load % 220 load % 800 load %
44 FZ1 = abs(FZ1);
45 L1 = FZ1 < 610 | FZ1 > 722 ; L2 = FZ1 < 942 | FZ1 > 1195 ; L3 = FZ1 < 190 | FZ1 > 240 ; L4 =
    FZ1 < 810 | FZ1 > 930 ;
46 FXG1 = FX1; FXG2 = FX1; FXG3 = FX1; FXG4 = FX1;
47 FXG1(L1) = [] ; FXG2(L2) = [] ; FXG3(L3) = [] ; FXG4(L4) = [] ;
48 SLG1 = SL1; SLG2 = SL1; SLG3 = SL1; SLG4 = SL1;
49 SLG1(L1) = [] ; SLG2(L2) = [] ; SLG3(L3) = [] ; SLG4(L4) = [] ;
50
51 figure(2);
52 plot(SLG1, FXG1);
53 grid on;
54 hold on;
55 plot(SLG2, FXG2, '-r');
56 plot(SLG3, FXG3, '-k');
57 plot(SLG4, FXG4, '-m');
58 legend('F_z = 670 N', 'F_z = 950 N', 'F_z = 220', 'F_z = 800 N', 'Location', 'southeast');
59 title('F_x vs \kappa');
60
61
62 FX2(L1)= [] ; FX3(L1) = [] ;
63 SL2(L1) = [] ; SL3(L1) = [] ;
64
65 figure(3)
```

```

66 plot(SLG1, FXG1, '-r'); % alpha 0 deg
67 grid on;
68 hold on;
69 plot(SL2, FX2, '-k');% alpha 3 deg
70 plot(SL3, FX3) % alpha 6 deg
71 legend('alpha = 0 deg', 'alpha = 3 deg', 'alpha = 6 deg', 'Location', 'southeast');
72 title('longitudinal tire force as function of \alpha');

```

A.2 Exercise 2.2

```

1 %% Execute the first fitting
2
3 % 1. Select the data alpha = 0, gamma = 0, Fz=Fz.nom
4 RDP = P < 78.000 | abs(SA)> 0.04 | IA > 0.04 | abs(FZ) < 810 | abs(FZ) > 930; % select rows to
   delete
5 P1 = P; P1(RDP) = []; % filter the pressure
6 FZF = abs(FZ); FZF(RDP) =[];% filter FZ0
7 SLF = SL; SLF(RDP) = [];% filter k
8 FXF = FX; FXF(RDP) =[];%filter fx
9
10 % 2. Write Pacejka MC and fit the data
11
12 X0 = zeros(1,7);
13 X0 = [0.0006 0.0001 0.0002 0.0001 0.0001 0.0005 0.00008];
14 IA0 = 0; FZ0 = -890;
15
16
17 Xop1 = fmincon(@(X) resid_pure_Fx(X,FXF,SLF,IA0, FZF),X0, [],[],[],[],[],[]);
18
19
20 FxF0F1 = FxP(Xop1, FXF,SLF,IA0, FZF);
21
22 figure(4);
23 plot(SLF, FXF, '-r');
24 hold on
25 grid on
26 plot(SLF, FxF0F1, '-b');
27 legend('raw data', 'fitted data', 'Location', 'southeast');
28 title('First Fitting');
29
30
31 % 3. Get the parameter
32 %% Execute the second fitting
33 % 1. Select the data for Fz=change and gamma = 0
34
35 RDP2 = P < 78.000 | abs(SA)> 0.04 | IA > 0.04 | abs(FZ) < 610 | abs(FZ) > 722 ; % select rows to
   delete
36 P2 = P; P2(RDP2) = []; % filter the pressure
37 FZF2 = abs(FZ); FZF2(RDP2) =[];% filter FZ0
38 SLF2 = SL; SLF2(RDP2) = [];% filter k
39 FXF2 = FX; FXF2(RDP2) =[];%filter fx
40 % second FZ parameters -----
41
42 RDP3 = P < 78.000 | abs(SA)> 0.04 | IA > 0.04 | abs(FZ) < 190 | abs(FZ) > 240 ; % select rows to
   delete
43 P3 = P; P3(RDP3) = []; % filter the pressure
44 FZF3 = abs(FZ); FZF3(RDP3) =[];% filter FZ0
45 SLF3 = SL; SLF3(RDP3) = [];% filter k
46 FXF3 = FX; FXF3(RDP3) =[];%filter fx
47
48 % third FZ parameters -----
49
50
51 RDP4 = P < 78.000 | abs(SA)> 0.04 | IA > 0.04 | abs(FZ) < 942 | abs(FZ) > 1195 ; % select rows to
   delete
52 P4 = P; P4(RDP4) = []; % filter the pressure
53 FZF4 = abs(FZ); FZF4(RDP4) =[];% filter FZ0
54 SLF4 = SL; SLF4(RDP4) = [];% filter k
55 FXF4 = FX; FXF4(RDP4) =[];%filter fx
56
57

```

```

58 X2F = zeros(1,7);
59 X2F= [0.01 0.05 0.006 0.008 0.002 0.004 0.002];
60 FXM = [FXF FXF2(1:486) FXF3(1:486) FXF4(1:486)];
61 FZM = [FZF FZF(1:486) FZF3(1:486) FZF4(1:486) ];
62 SLFM = [SLF SLF2(1:486) SLF3(1:486) SLF4(1:486) ];
63
64 Xop2 = fmincon(@(X) resid_pure_Fx_varFz(X,FXM,SLFM,IA0, FZM, Xop1),X2F, [], [], [], [], []);
65
66 Xplot2 = horzcat(Xop1,Xop2)
67
68 %FxP2(X,FX,K,gamma, Fz)
69 Fx0F2 = FxP2(Xplot2, FXM,SLFM,IA0, FZM);
70
71 figure(5);
72 plot(SLFM, FXM, 'b');
73 hold on
74 grid on
75 plot(SLFM, Fx0F2, 'k');
76
77 %Fx0F2 = resid_pure_Fx_varFz(X,FXM,SLFM,IA0, FZM, Xop1, a)
78 % 2. Fit the data using the previous parameters
79 % 4. Get the new parameters
80 %% Execute the third fitting
81 % 1. Select the data for Fz=-890 and gamma = 0
82 RDP31 = P < 78.000 | abs(SA)> 0.04 | IA > 0.04 | abs(FZ) < 810 | abs(FZ) > 930 ; % select rows to
     delete
83 P5 = P; P5(RDP31) = []; % filter the pressure
84 FzTF = -890;
85 SLP31 = SL; SLP31(RDP31) = []; % filter k
86 FXF31 = FX; FXF31(RDP31) =[]; %filter fx
87 IA31 = IA; IA31(RDP31) = [];
88
89 % ----- gamma = 2 -----
90 RDP32 = P < 78.000 | abs(SA)> 0.04 | IA < 1.98 | IA > 2.06 | abs(FZ) < 810 | abs(FZ) > 930 ; %
     select rows to delete
91 P6 = P; P6(RDP32) = []; % filter the pressure
92
93 SLP32 = SL; SLP32(RDP32) = []; % filter k
94 FXF32 = FX; FXF32(RDP32) =[]; %filter fx
95 IA32 = IA; IA32(RDP32) = [];
96
97 % ----- gamma = 4 -----
98
99 RDP33 = P < 78.000 | abs(SA)> 0.04 | IA < 3.95 | IA > 4.05 | abs(FZ) < 810 | abs(FZ) > 930 ; %
     select rows to delete
100 P7 = P; P7(RDP33) = []; % filter the pressure
101 FzTF = -890;
102 SLP33 = SL; SLP33(RDP33) = []; % filter k
103 FXF33 = FX; FXF33(RDP33) =[]; %filter fx
104 IA33 = IA; IA33(RDP33) = [];
105
106
107 X3F = zeros(1,1);
108 X3F(1) = 0.02;
109
110 FXM3 = [FXF31; FXF32; FXF33];
111 SLFM3 =[SLP31; SLP32; SLP33];
112 IAM3 =[IA31; IA32; IA33];
113
114
115 Xop3 = fmincon(@(X) resid_pure_Fx_varCamber(X,FXM3,SLFM3,IAM3, FzTF, Xop1, Xop2),X3F, [], [], []);
116
117
118
119 Xopt = [Xop1 Xop2 Xop3];

```

A.2.1 function resid_pure_Fx

```

1 function res = resid_pure_Fx(X,FX,K,gamma, Fz)
2
```

```

3 % _____
4 %% Compute the residuals - least squares approach - to fit the Fx curve
5 % with Fz=Fz_nom, IA=0. Pacejka 1996 Magic Formula
6 %
7
8 % Define MF coefficients
9
10 Fz0 = -890;
11 dfz = Fz / Fz0 -1;
12
13 pHx1 = X(1); pCx1 = X(2);
14 pDx1 = X(3); pKx1 = X(4);
15 pEx1 = X(5); pEx4 = X(6);
16 pVx1 = X(7); pDx2 = 0;
17 pDx3 = 0; pEx2 = 0;
18 pEx3 = 0; pKx2 = 0;
19 pKx3 = 0; pHx2 = 0;
20 pVx2 = 0;
21
22
23 SHx = pHx1 + pHx2*dfz;
24 Kx = K + SHx; % eq 1
25 Cx = pCx1; % eq 2
26 Mux = (pDx1+pDx2*dfz)*(1-pDx3 *gamma^2) ;
27 Dx = Mux .* Fz;% eq 3
28 Kxk = Fz .* (pKx1 + pKx2.*dfz).*exp(-pKx3.*dfz); % eq 4
29 Ex = (pEx1 + pEx2.*dfz) .* (1-pEx4.*sign(Kx)); % eq 5
30 Bx = Kxk ./ (Cx.*Dx);
31 Svx = Fz .*(pVx1+pVx2.*dfz); % eq 6
32
33
34 Fx0 = Dx.*sin(Cx.*atan(Bx.*Kx- Ex.* (Bx.*Kx-atan(Bx.*Kx)) ))+Svx;
35
36 % Compute the residuals
37
38 res = sum((FX - Fx0).^2)/sum(FX.^2);
39 end

```

A.2.2 function FXP

```

1 function Fplotting = FxP(X,FX,K,gamma, Fz)
2 %
3 % _____
4 %% Compute the residuals - least squares approach - to fit the Fx curve
5 % with Fz=Fz_nom, IA=0. Pacejka 1996 Magic Formula
6 %
7
8 % Define MF coefficients
9
10 Fz0 = -890;
11 dfz = Fz / Fz0 -1;
12
13 pHx1 = X(1); pCx1 = X(2);
14 pDx1 = X(3); pKx1 = X(4);
15 pEx1 = X(5); pEx4 = X(6);
16 pVx1 = X(7); pDx2 = 0;
17 pDx3 = 0; pEx2 = 0;
18 pEx3 = 0; pKx2 = 0;
19 pKx3 = 0; pHx2 = 0;
20 pVx2 = 0;
21
22
23 SHx = pHx1 + pHx2*dfz;
24 Kx = K + SHx; % eq 1
25 Cx = pCx1; % eq 2
26 Mux = (pDx1+pDx2*dfz)*(1-pDx3 *gamma^2) ;
27 Dx = Mux .* Fz;% eq 3
28 Kxk = Fz .* (pKx1 + pKx2.*dfz).*exp(-pKx3.*dfz); % eq 4
29 Ex = (pEx1 + pEx2.*dfz) .* (1-pEx4.*sign(Kx)); % eq 5
30 Bx = Kxk ./ (Cx.*Dx);
31 Svx = Fz .*(pVx1+pVx2.*dfz); % eq 6

```

```

32
33
34 Fx0 = Dx.* sin (Cx.* atan (Bx.*Kx- Ex.* (Bx.*Kx-atan (Bx.*Kx) )) )+Svx;
35
36 % Compute the residuals
37 Fplotting = Fx0;
38 end

```

A.2.3 function resid_pure_Fx_varCamber

```

1 function res = resid_pure_Fx_varCamber(X,FX, K, gamma, Fz, X1, X2)
2
3 % _____
4 %% Compute the residuals - least squares approach - to fit the Fx curve
5 % with Fz=Fz_nom, IA=0. Pacejka 1996 Magic Formula
6 % _____
7
8 % Define MF coefficients
9
10 Fz0 = -890;
11 dfz = Fz / Fz0 -1;
12
13 pHx1 = X1(1); pCx1 = X1(2);
14 pDx1 = X1(3); pKx1 = X1(4);
15 pEx1 = X1(5); pEx4 = X1(6);
16 pVx1 = X1(7); pDx2 = X2(1);
17 pDx3 = X2(2); pEx2 = X2(3);
18 pEx3 = X1(1); pKx2 = X2(4);
19 pKx3 = X2(5); pHx2 = X2(6);
20 pVx2 = X2(7);
21
22
23 SHx = pHx1 + pHx2*dfz;
24 Kx = K + SHx; % eq 1
25 Cx = pCx1; % eq 2
26 Mux = (pDx1+pDx2*dfz)*(1-pDx3 .* gamma.^2) ;
27 Dx = Mux .* Fz;% eq 3
28 Kxk = Fz .* (pKx1 + pKx2.*dfz).*exp(-pKx3.*dfz); % eq 4
29 Ex = (pEx1 + pEx2.*dfz) .* (1-pEx4.*sign(Kx)); % eq 5
30 Bx = Kxk ./ (Cx.*Dx);
31 Svx = Fz.* (pVx1+pVx2.*dfz); % eq 6
32
33
34 Fx0 = Dx.* sin (Cx.* atan (Bx.*Kx- Ex.* (Bx.*Kx-atan (Bx.*Kx) )) )+Svx;
35
36 % Compute the residuals
37 res = sum((FX - Fx0).^2)/sum(FX.^2);
38
39 end

```

A.2.4 function resid_pure_Fx_varFz

```

1 function res = resid_pure_Fx_varFz(X,FX,K,gamma, Fz, X1)
2
3 % _____
4 %% Compute the residuals - least squares approach - to fit the Fx curve
5 % with Fz=Fz_nom, IA=0. Pacejka 1996 Magic Formula
6 % _____
7
8 % Define MF coefficients
9 FZ1 =Fz(1:486);
10 FZ2 =Fz(487:1022);
11 FZ3 =Fz(1023:1533);
12 FZ4 =Fz(1534:2071);
13
14 FX1 =FX(1:486);
15 FX2 =FX(487:1022);
16 FX3 =FX(1023:1533);
17 FX4 =FX(1534:2071);
18
19 KF1 = K(1:486);

```

```

20 KF2 = K(1:486);
21 KF3 = K(1:486);
22 KF4 = K(1:486);
23
24
25 Fz0 = -890;
26 dfz = Fz ./ Fz0 -1;
27
28 pHx1 = X1(1); pCx1 = X1(2);
29 pDx1 = X1(3); pKx1 = X1(4);
30 pEx1 = X1(5); pEx4 = X1(6);
31 pVx1 = X1(7); pDx2 = X(1);
32 pDx3 = X(2); pEx2 = X(3);
33 pEx3 = 0; pKx2 = X(4);
34 pKx3 = X(5); pHx2 = X(6);
35 pVx2 = X(7);
36
37
38 SHx = pHx1 + pHx2*dfz;
39 Kx = K + SHx; % eq 1
40 Cx = pCx1; % eq 2
41 Mux = (pDx1+pDx2*dfz)*(1-pDx3 *gamma^2) ;
42 Dx = Mux .* Fz; % eq 3
43 Kxk = Fz .* (pKx1 + pKx2.* dfz).*exp(-pKx3.* dfz); % eq 4
44 Ex = (pEx1 + pEx2.* dfz) .* (1-pEx4.* sign(Kx)); % eq 5
45 Bx = Kxk ./ (Cx.*Dx);
46 Svx = Fz.* (pVx1+pVx2.* dfz); % eq 6
47
48
49 Fx0 = Dx.* sin (Cx.* atan (Bx.* Kx- Ex.* (Bx.* Kx-atan (Bx.* Kx)))))+Svx;
50
51 % Compute the residuals
52 res = sum((FX - Fx0).^2)/sum(FX.^2);
53
54 end

```