Vehicle Dynamics report

Francesco Pegonzi

A.Y. 2023-2024

1 Introduction

The goal of this project is to take telemetry data of one lap of a F1 race car and process them using appropriate kinematic and dynamic models to achieve multiple tasks.

In particular it is required to:

- Load data from Excel, convert them into SI units and flip the sign if necessary;
- Plot signals;
- Filter and differetiate the yaw rate;
- Reconstruct the trajectory of the vehicle center of mass;
- Evaluate aerodynamics loads;
- Estimate the physical grip;
- Evaluate the lateral forces Y_1 and Y_2;
- Evaluate the yaw moment N;
- Obtain the g-g plot;
- Draw the fixed centrodes of all curves;
- Draw the moving centrode of one curve in three positions.
- Finding power-limited curves and grip-limited curves;

The telemetry data are provided in an *Excel* file, and *Matlab* is used to achieve the tasks.

2 Loading data

To load data from *Excel* to *Matlab* the command **xlsread()** is used, in this way:

```
% Defining file path
file_path = 'telemetrie_2012_per_2023.xls';

% Defining the variables where the data are saved
[num_data] = xlsread(file_path);
```

Bear in mind to put the *Excel* file in the same folder as the *Matlab* script.

In this way, we have a matrix in which each column corresponds to a specific type of data, and each row to a specific time sample. After that it is possible to create specific arrays containing specific columns of **num_data**, to make the script more readable.

```
time = num_data(:,1);                     % [time] = s
dist = num_data(:,2);                     % [dist] = m
speed = num_data(:,3);                    % [speed] = km/h
ax = num_data(:,4);                       % [ax] = m/s^2/g
ay = num_data(:,5);                       % [ay] = m/s^2/g
airspeed = num_data(:,6);                 % [airspeed] = km/h
sideslip_front = num_data(:,7);           % [sideslip_front] = deg
sideslip_rear =  num_data(:,8);           % [sideslip_rear] = deg
omega_z = num_data(:,9);                  % [omega_z_vettura] = deg/s
```

```
massa_vettura = num_data(:,10);         % [massa_vettura] = kg
farf = num_data(:,11);                  % [farf] = %
steer = num_data(:,12);                 % [steer] = deg
p_brake = num_data(:,13);               % [p_brake] = psi
carreggiata_front = num_data(:,14);     % [carreggiata_front] = m
carreggiata_rear = num_data(:,15);      % [carreggiata_rear] = m
weight_dist = num_data(:,16);           % [weight_dist] = fraction
Jxx = num_data(:,17);                   % [Jxx] = kg*m^2
Jyy = num_data(:,18);                   % [Jyy] = kg*m^2
Jzz = num_data(:,19);                   % [Jzz] = kg*m^2
Jzx = num_data(:,20);                   % [Jzx] = kg*m^2
passo_vettura = num_data(:,21);         % [passo_vettura] = m
cz_front = num_data(:,22);              % [cz_front] = m^2
cz_rear = num_data(:,23);               % [cz_rear] = m^2
air_dens = num_data(:,24);              % [air_dens] = kg/
```

2.1 SI conversion

With common sense considerations, some data were converted into the SI units:

```
speed = speed * 1e3/3600;
airspeed = airspeed * 1e3/3600;
ax = ax * g;
ay = ay * g;
omega_z = - omega_z * pi/180;
sideslip_front = sideslip_front*pi/180;
sideslip_rear = sideslip_rear*pi/180;
steer = steer*pi/180;
p_brake = p_brake*6894.76;
```

Then, it is necessary to change the sign of the yaw rate **omega_z** to make it consistent with the steering angle.

3 Plotting signals

3.1 Speed

Two speed signals were plotted:

- speed: This is the velocity of the vehicle along the longitudinal axis of the vehicle frame;
- airspeed: This is the relative velocity between the airflow and the vehicle.

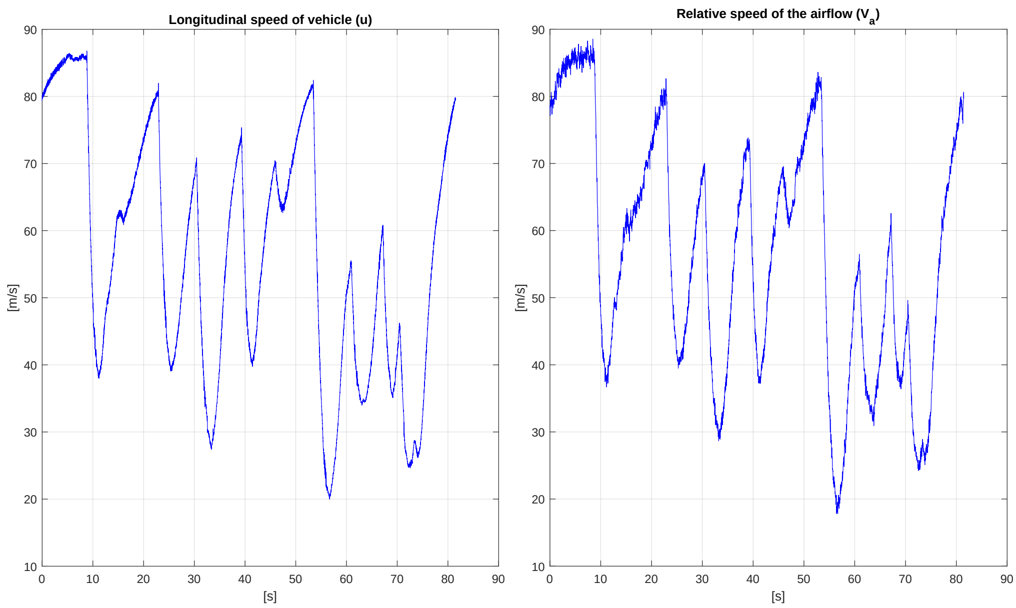

Figure 3.1: Speed plotting

3.2 Accelerations

Two accelerations signals were plotted:

- a_x: This is the longitudinal acceleration of the vehicle;
- a_y: This is the lateral acceleration of the vehicle.

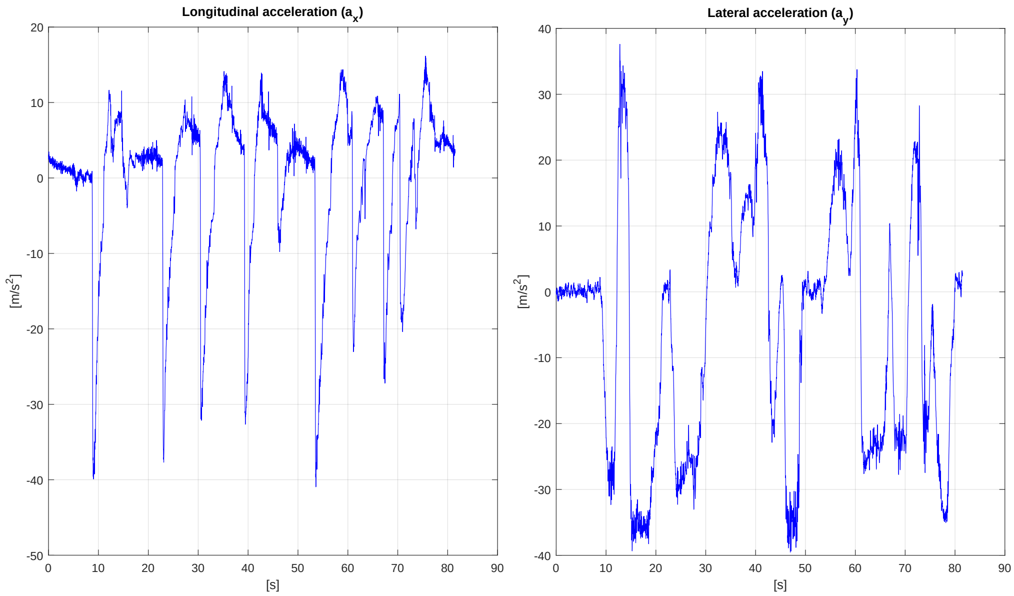

Figure 3.2: Accelerations plotting

3.3 Angles

Three angles signals were plotted:

- sideslip_front: This is the front slip angle of the wheels α_f;
- sideslip_rear: This id the rear slip angle of the wheels α_r;
- steer: This is the steering angle.

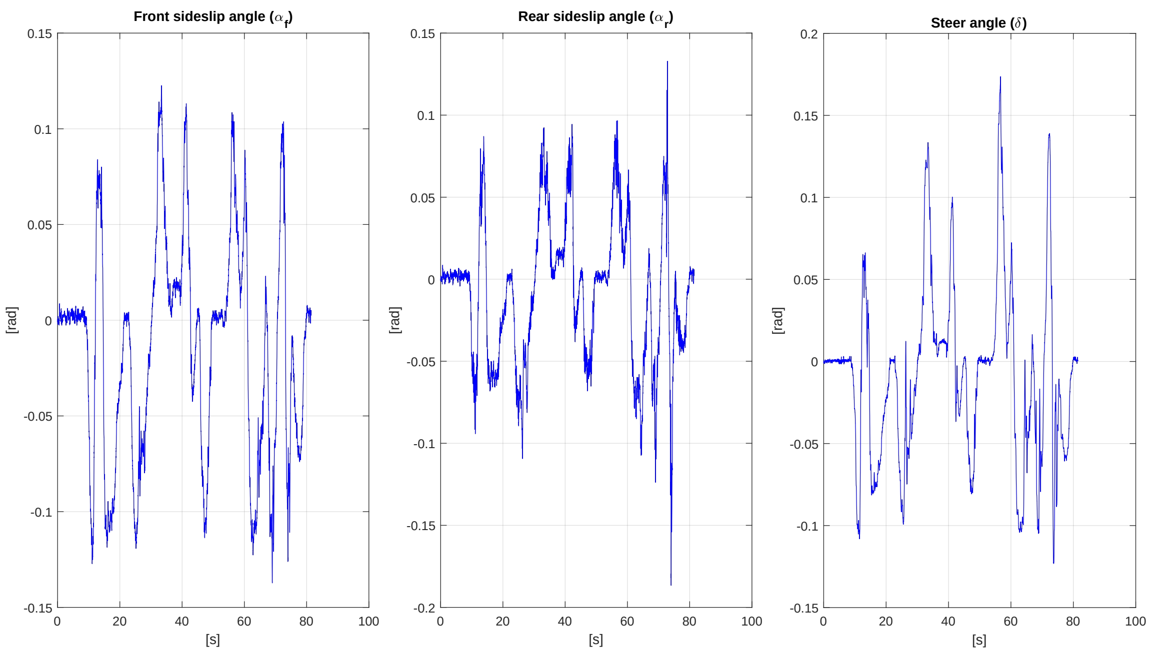

Figure 3.3: Angles plotting

3.4 Brake pressure and farf

Two signals were plotted:

- p_brake: This is the pressure of the oil in the braking system.
- farf: This is the opening of the valve of the motor, expressed as a percentage.

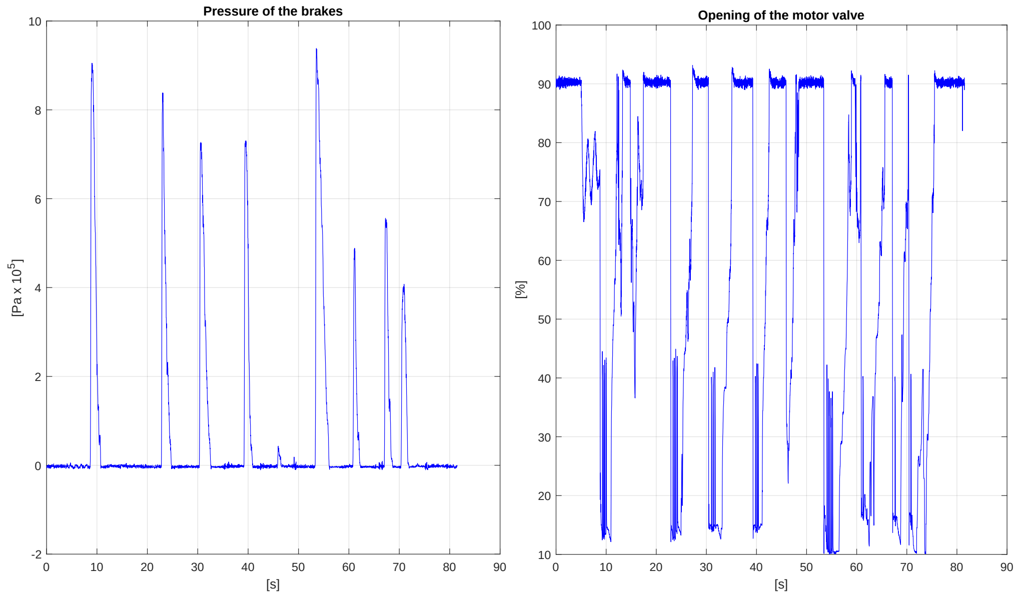

Figure 3.4

3.5 Yaw rate, filtered yaw rate and differentiated yaw rate

Three signals were plotted:

- omega_z: This is the yaw rate of the vehicle;
- omega_z _f: This is the filtered yaw rate, obtained from the original ω_z using the command **smoothdata()**;
- d_omega_z_f: This is the differentiated yaw rate, obtained from the filtered ω_z using forward Euler.

The portion of the script related to the filtering and differentiation is reported over here:

```
% Filtering omega_z
omega_z_f = smoothdata(omega_z, "loess","SmoothingFactor",0.25);

% Differentiating omega_z
d_omega_z_f = zeros(n,1);
for i=1:n-1
    d_omega_z_f(i) =  ( omega_z_f(i+1) - omega_z_f(i) ) / dt;
end
d_omega_z_f(n) = d_omega_z_f(n-1);
```

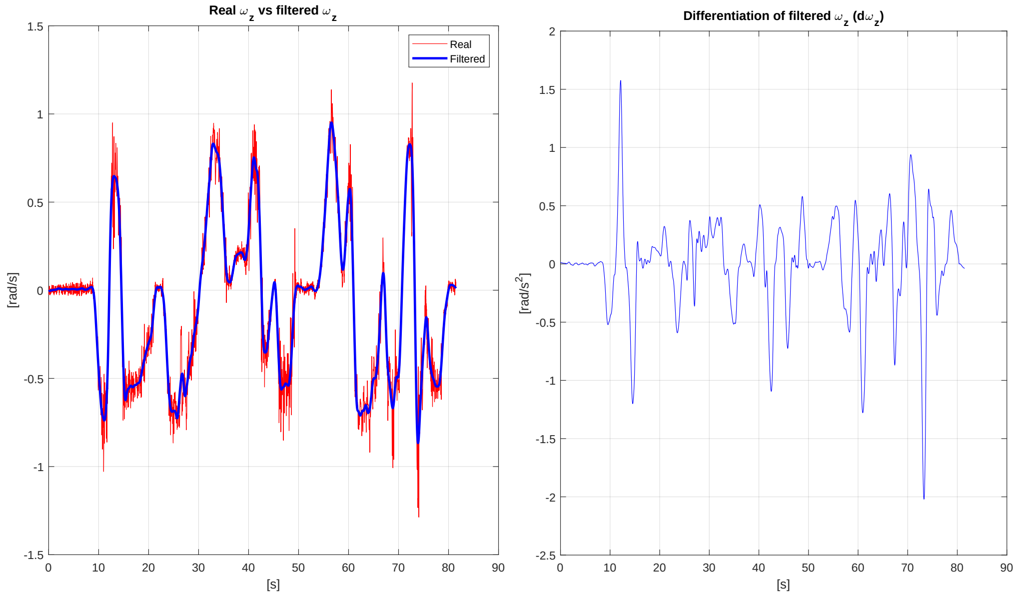

Figure 3.5: Yaw rate, filtered yaw rate and differentiated yaw rate

4 Trajectory reconstruction of the center of mass

The goal of this section is the reconstruction of the trajectory of the center of mass, using relevant telemetry data.

4.1 Reference frames

For the reconstruction of the trajectory of the center of mass two reference frames are used:

- $S_0 = (x_0, y_0, z_0; O)$: This is the fixed or inertial reference frame;
- $S = (x, y, z; G)$: This is the body fixed reference frame.

As we can see in figure 4.1, the relationship between the orientation of the vehicle frame and the fixed frame are:

$$\mathbf{i} = -\mathbf{i_0} \cdot cos(\psi) - \mathbf{j_0} \cdot sin(\psi) \tag{4.1}$$
$$\mathbf{j} = \mathbf{i_0} \cdot sin(\psi) - \mathbf{j_0} \cdot cos(\psi) \tag{4.2}$$

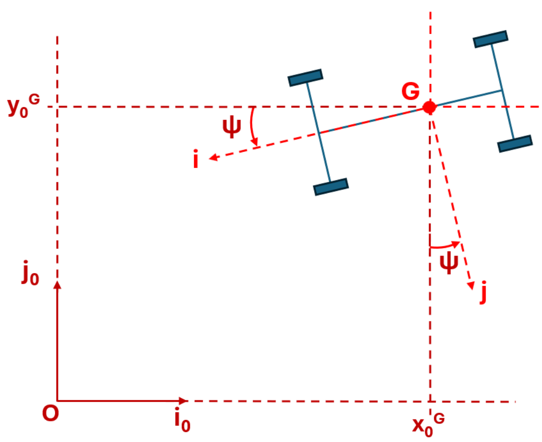

Figure 4.1: Frames used for the reconstruction of G

4.2 Kinematic model

The model used to the reconstruction of the trajectory of G is the single track model (figure 4.2).

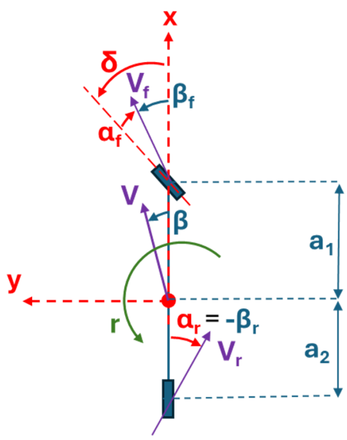

Figure 4.2: Single track model

From the telemetry data we can get:

- speed = u;
- sideslip_front = α_f;
- sideslip_rear = α_r;
- omega_z = r;

β_f and β_r can be written in function of $\delta, \alpha_f, \alpha_r$ as:

$$\beta_f = \delta - \alpha_f \tag{4.3}$$
$$\beta_r = -\alpha_r \tag{4.4}$$

V is the velocity of G and it can be written in vehicle frame as:

$$\mathbf{V} = u \cdot \mathbf{i} + v \cdot \mathbf{j} \tag{4.5}$$

From the fundamental formula of the planar kinematics we can compute **v** using the front point and the rear point, so we obtain for a generic instant k:

$$v_1(k) = tan(\beta_f(k)) \cdot u(k) - r(k) \cdot a1 \tag{4.6}$$
$$v_2(k) = tan(\beta_r(k)) \cdot u(k) + r(k) \cdot a2 \tag{4.7}$$

Ideally the value of **v1** (which is obtained from the equation 4.6) and **v2** (which is obtained from the equation 4.7), should give the same results. however, due to the measurement noise we obtain different values, so that the definitive value is obtained with an average of the two v:

$$v(k) = 0.5 * (v_1(k) + v_2(k)) \tag{4.8}$$

Knowing the value of **u**, **v** and **r**, we can reconstruct the trajectory of G:

$$x_0^G(t) = x_0^G(0) + \int_0^t (-u(\tau) \cdot cos(\psi(\tau) + v(\tau) \cdot sin(\psi(\tau)))d\tau \tag{4.9}$$

$$y_0^G(t) = y_0^G(0) + \int_0^t (-u(\tau) \cdot cos(\psi(\tau) - v(\tau) \cdot sin(\psi(\tau)))d\tau \tag{4.10}$$

$$\psi(t) = \psi(0) + \int_0^t r(\tau)d\tau \tag{4.11}$$

Having discrete sample of measurement, we have to convert the the equations 4.9, 4.10 and 4.14 in discrete integrals. Using forward Euler integration, we obtain:

$$x_0^G(k+1) = x_0^G(k) + (-u(k) \cdot cos(\psi(k) + v(k) \cdot sin(\psi(k))) \cdot T \tag{4.12}$$
$$y_0^G(k+1) = y_0^G(k) + (-u(k) \cdot cos(\psi(k) - v(k) \cdot sin(\psi(k))) \cdot T \tag{4.13}$$
$$\psi(k+1) = \psi(k) + r(k) \cdot T \tag{4.14}$$

Where T is the sample time.

4.2.1 Error propagation

Due to measurements noises, if we only do the forward integration with Euler, setting the starting values at k = 0, we obtain a trajectory affected by error propagation, especially in the final phase of the estimation. One of the possible solutions is to do a backward integration, setting the final value at k = 8150. After that, we can do a weighted average defining two weight vectors that allow to obtain a good estimation of the trajectory of the center of mass.

4.3 Implementation

Later on we have the implementation on *Matlab* of the forward and backward integration, with the weighted average and the weight vectors definitions:

```
% Forward integration
v1_f = zeros(n,1);
v2_f = zeros(n,1);
v_f = zeros(n,1);
xg_f = zeros(n,1);
yg_f = zeros(n,1);
yaw_f = zeros(n,1);

for i=1:n-1
    v1_f(i) = tan(beta_f(i))*speed(i) - omega_z(i)*a1;
    v2_f(i) = tan(beta_r(i))*speed(i) + omega_z(i)*a2;

    v_f(i) = 0.5*( v1_f(i) + v2_f(i) );

    xg_f(i+1) = xg_f(i) - ( speed(i)*cos( yaw_f(i) ) - v_f(i)*sin( yaw_f(i) ) )*dt;
    yg_f(i+1) = yg_f(i) - ( speed(i)*sin( yaw_f(i) ) + v_f(i)*cos( yaw_f(i) ) )*dt;

    yaw_f(i+1) = yaw_f(i) + 1/2*( omega_z(i+1) + omega_z(i) )*dt;
    yaw_f(i+1) = atan2( sin( yaw_f(i+1) ), cos( yaw_f(i+1) ) );
end

% Backward integration
v1_b = zeros(n,1);
v2_b = zeros(n,1);
v_b = zeros(n,1);
xg_b = zeros(n,1);
yg_b = zeros(n,1);
yaw_b = zeros(n,1);

for i=n:-1:2
    v1_b(i) = tan(beta_f(i))*speed(i) - omega_z(i)*a1;
    v2_b(i) = tan(beta_r(i))*speed(i) + omega_z(i)*a2;

    v_b(i) = 0.5*( v1_b(i) + v2_b(i) );

    xg_b(i-1) = xg_b(i) + ( speed(i)*cos( yaw_b(i) ) - v_b(i)*sin( yaw_b(i) ) )*dt;
    yg_b(i-1) = yg_b(i) + ( speed(i)*sin( yaw_b(i) ) + v_b(i)*cos( yaw_b(i) ) )*dt;

    yaw_b(i-1) = yaw_b(i) - ( omega_z(i) )*dt;
    yaw_b(i-1) = atan2( sin( yaw_b(i-1) ), cos( yaw_b(i-1) ) );
end

% Defining weight vectors
wf = linspace(1,0,n)';          % Forward weight vector
wb = linspace(0,1,n)';          % Backward weight vector

% G trajectory
xg = wf.*xg_f + wb.*xg_b;
yg = wf.*yg_f + wb.*yg_b;
```

4.4 Plotted trajectory

As a result we obtain:

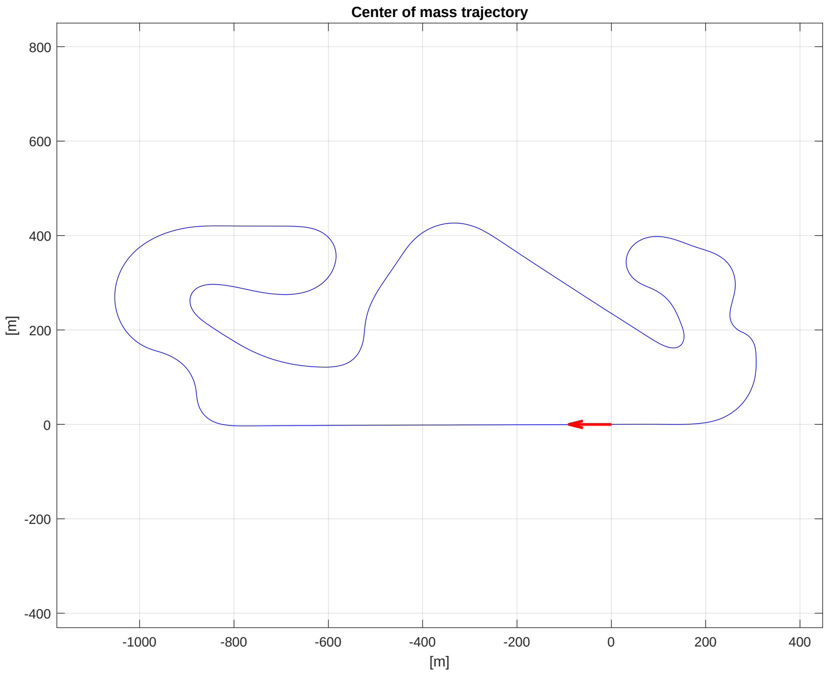

Figure 4.3: Trajectory reconstruction of the center of mass

5 Aerodynamic loads evaluation

The aerodynamics loads, represented in figure 5.2, are defined in the following way:

- X_a: Dragforce;
- Z_1^a: Front downforce;
- Z_2^a: Rear downforce.

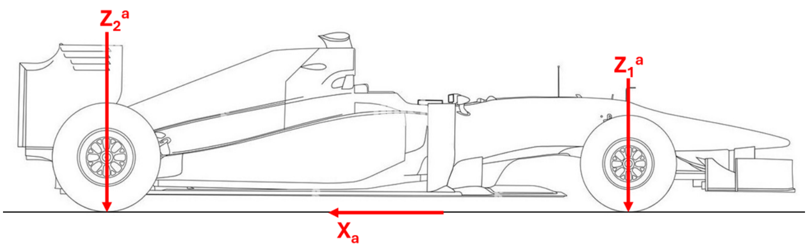

Figure 5.1: Aerodynamic loads

At a generic time t the formulas of that loads are defined as:

$$X_a(t) = \frac{1}{2}\rho_a V_a^2(t) S_a C_x(t) = \xi(t) V_a^2(t) \tag{5.1}$$

$$Z_1^a(t) = \frac{1}{2}\rho_a V_a^2(t) S_a C_{z1}(t) = \zeta_1(t) V_a^2(t) \tag{5.2}$$

$$Z_2^a(t) = \frac{1}{2}\rho_a V_a^2(t) S_a C_{z2}(t) = \zeta_2(t) V_a^2(t) \tag{5.3}$$

With:

- ρ_a: Air density;
- $V_a(t)$: Magnitude of the relative velocity between vehicle and air;

- S_a: Area of the vehicle frontal projection;
- $C_x(t)$: Drag coefficient;
- $C_{z1}(t)$: Front downforce coefficient;
- $C_{z2}(t)$: Rear downforce coefficient.

For the computation at any time of the aerodynamic loads, the following data must be taken from the telemetry:

- air_dens $= \rho_a$;
- airspeed $= V_a$;
- cx_tot $= S_a \cdot C_x$;
- cz_front $= S_a \cdot C_{z1}$;
- cz_rear $= S_a \cdot C_{z2}$.

5.1 Implementation

The estimation of aerodynamic loads at any time, can be obtained with a for cicle:

```
%% Aerodynamic loads estimation
rho = air_dens(1);
Xa = zeros(n,1);
Za1 = zeros(n,1);
Za2 = zeros(n,1);

for i=1:n
    Xa(i) = 0.5*rho*airspeed(i)^2*cx_tot(i);
    Za1(i) = 0.5*rho*airspeed(i)^2*cz_front(i);
    Za2(i) = 0.5*rho*airspeed(i)^2*cz_rear(i);
end
```

5.2 Results

The following loads are obtained:

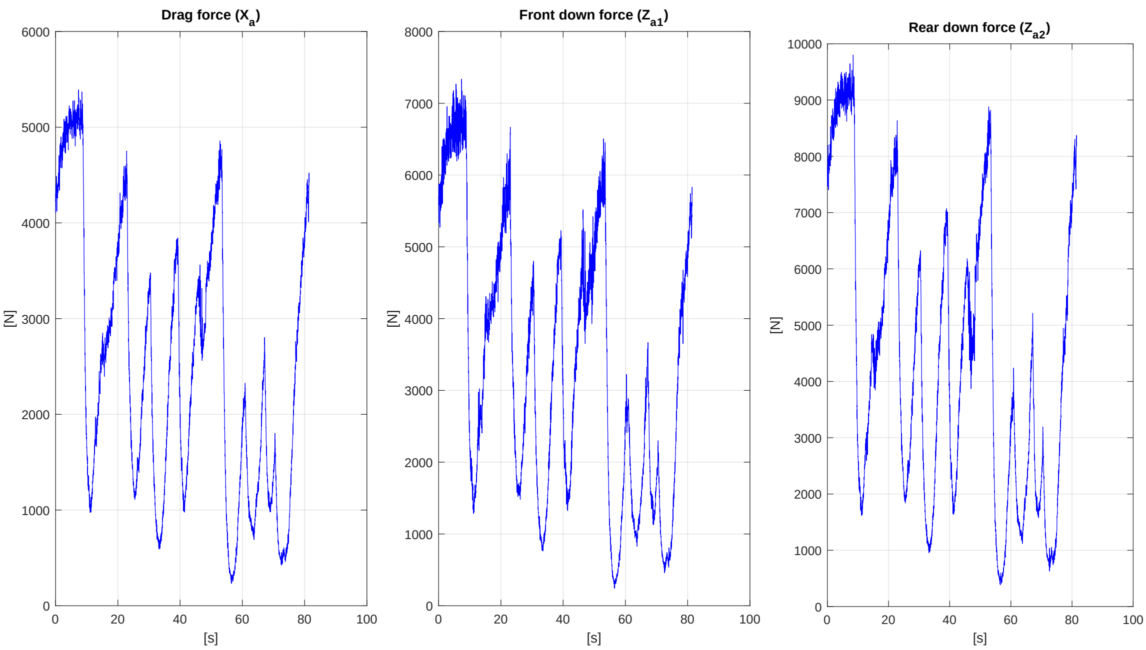

Figure 5.2: Aerodynamics loads

6 Physical grip estimation

For the estimation of the physical grip μ, it's necessary to analyse the loads acting on the vehicle at the time of maximum deceleration.

6.1 Braking performance

It is assumed that the vehicle have maximum deceleration when it is going on a flat, straight road with uniform grip. A scheme of the braking condition is represented in figure 6.1.

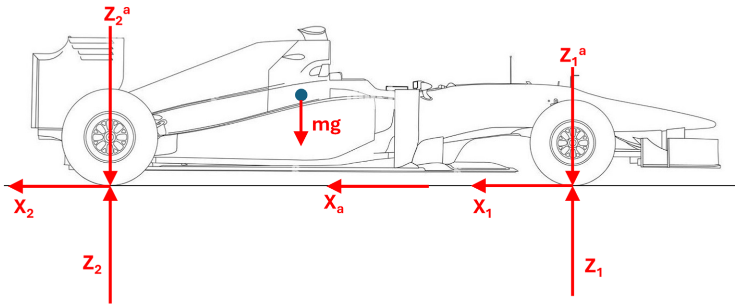

Figure 6.1: Braking performance

In addition, it is assumed that at the maximum deceleration the vehicle is in the condition of best braking performance, that is:

$$X_1 = \mu Z_1 \tag{6.1}$$

$$X_2 = \mu Z_2 \tag{6.2}$$

Therefore, from the equilibrium equations, the physical grip μ is obtained as:

$$\mu = \frac{|a_{min}| - \frac{X_a}{m}}{g + \frac{Z_1^a + Z_2^a}{m}} \tag{6.3}$$

With the time variant quantities computed at the time wich a_x is minimun.

6.2 Implementation

For the physical grip estimation, it is first necessary a minimum search algorithm, after that μ can be computed as in the equation 6.3:

```
%% Physical grip estimation

% Minimum ax search
ax_min = 0;                         % minimum longitudinal acceleration
n_min = 0;                          % sample to wich we have ax_min
for i=1:n
    if ax(i) < ax_min
        ax_min = ax(i);
        n_min = i;
    end
end

% Grip estimation (mu)
mu = ( abs(ax_min) - ( Xa(n_min)/massa_vettura(n_min) ) ) / ( g + ( Za1(n_min) ...
    + Za2(n_min) )/massa_vettura(n_min) );
```

6.3 Results

Assuming that the pilot is braking at the limit braking conditions, the physical grip results $\boldsymbol{\mu = 1.17}$.

It is a smaller value than usual, this could be due to the fact that the pilot isn't pushing the car to his braking limit.

7 Lateral forces evaluation

The goal of this section is the evaluation of the lateral forces Y_1 and Y_2

7.1 Single track model

A first approach on the evaluation of Y_1 and Y_2 can be done using the single track model, a scheme of the longitudinal and lateral tire forces acting on the single track model is represented in figure 7.1.

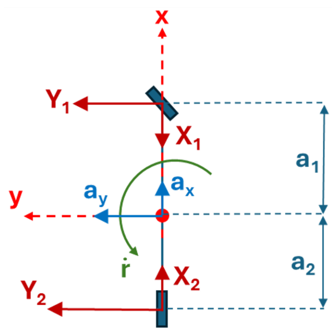

Figure 7.1: Longitudinal and lateral tire forces in the single track model

For the evaluation at any time of the lateral forces, two equation are involved; the first cardinal equation of the dynamics along the y axis 7.1 and the second cardinal equation of the dynamics along the z axis 7.2.

$$Y_1(t) + Y_2(t) = m(t) \cdot a_y(t) \tag{7.1}$$

$$J_{zz} \cdot \dot{r}(t) = Y_1(t) \cdot a_1 - Y_2(t) \cdot a_2 \tag{7.2}$$

Manipulating the equations 7.1 and 7.2, Y_1 and Y_2 can be obtained:

$$Y_1(t) = \frac{\dot{r}(t) \cdot J_{zz} + m(t) \cdot a_y(t) \cdot a2}{l} \tag{7.3}$$

$$Y_2(t) = m(t) \cdot a_y(t) - Y_1(t) \tag{7.4}$$

with $l = a_1 + a_1$.

For the estimation of that forces, the following data must be taken from the telemetry and from the previous computations:

- d_omega_z_f $= \dot{r}$, computed in section 3.5;
- J_zz;
- massa_vettura = m;
- a_y;
- passo_vettura $= l$.

7.1.1 Implementation

The evaluation of the lateral forces can be done with a for cicle, where the equations 7.3 and 7.20 are involved:

```
% Evaluating Y1 and Y2 using single track model
Y1 = zeros(n,1);
Y2 = zeros(n,1);

for i = 1:n
    Y1(i) = ( d_omega_z_f(i)*Jzz(1) + massa_vettura(i)*ay(i) ) / passo_vettura(1);
    Y2(i) = massa_vettura(i)*ay(i) - Y1(i);
end
```

7.2 Double track model

The single track model as limitations, and may not provide a good approximation of the lateral forces in a race car. This is due to the fact that race cars are characterised by rear wheel drive equipped with a limited slip differential. A scheme of the longitudinal and lateral tire forces acting on the double track model is represented in figure 7.2.

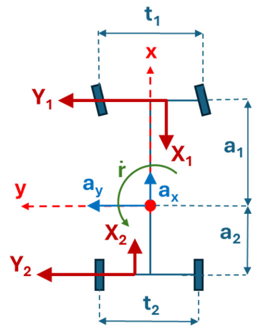

Figure 7.2: Longitudinal and lateral tire forces in the double track model

In that case, the equation 7.1 doesn't change, whereas the equation 7.2, becomes:

$$J_{zz} \cdot \dot{r}(t) = Y_1(t) \cdot a_1 - Y_2(t) \cdot a_2 + \Delta X_1(t) \cdot t_1 + \Delta X_2(t) \cdot t_2 \tag{7.5}$$

$$\Delta X_1 = \frac{X_{12} - X_{11}}{2} \tag{7.6}$$

$$\Delta X_2 = \frac{X_{22} - X_{21}}{2} \tag{7.7}$$

From the telemetry we can see that the steering angle is under $0.2rad$, therefore $\Delta X_1 \cdot t_1$ is negligible, whereas the contribution provided by ΔX_2, is due to the differential, and can't be neglected.

7.2.1 Limited slip differential

it is a 2 degree of freedom mechanism wich allows the two wheels to rotate at different speed. With that mechanism, The longitudinal forces of the rear wheels aren't the same and are related by the following equations:

$$X_{22} = \eta_h^{\zeta(t)} \cdot X_{21} \tag{7.8}$$

$$\zeta(t) = \frac{arctan(\chi \Delta\hat{\omega}(t))}{\pi/2} \tag{7.9}$$

$$\Delta\hat{\omega} = \omega_{22} - \omega_{21} \tag{7.10}$$

7.2.2 Rigid rear wheels hypothesis

If the double track model is wanted to be used, a first possible approach may be to assume rigid rear wheels, therefore the longitudinal velocities of the two rim of the rear wheels can be related to their angular velocities with the equations 7.11 7.12, defining r_r the rolling radius, which could be taken as $360mm$, referring to the datasheet of Pirelli racing wheels.

$$\omega_{21}(t) = \frac{u_{21}(t)}{r_r} \tag{7.11}$$

$$\omega_{22}(t) = \frac{u_{22}(t)}{r_r} \tag{7.12}$$

Where u_{21} and u_{22} can be computed from the fundamental formula of planar kinematics as:

$$u_{21}(t) = u(t) - r \cdot \frac{t_2}{2} \tag{7.13}$$

$$u_{22}(t) = u(t) + r \cdot \frac{t_2}{2} \tag{7.14}$$

At that point $\zeta(t)$ can be computed with the equation 7.9, assuming the time constant $\chi = 1000s$.

After that writing the first cardinal equation, along the x axis (eq. 7.15) and assuming X_1 negligible, due to small steering angles, X_2 can be obtained using equation 7.16.

$$m(t) \cdot a_x(t) = X_1(t) - X_2(t) - X_a(t) \tag{7.15}$$

$$X_2(t) = m(t) \cdot a_x(t) + X_a \tag{7.16}$$

Combining the limited slip differential equation 7.8, with the equation 7.16, assuming $\eta_h = 0.5$ and solving for X_{21} and X_{22} the following equations are obtained:

$$X_{21}(t) = \frac{X_2(t)}{\eta_h^{\zeta(t)} + 1} \tag{7.17}$$

$$X_{22}(t) = \eta_h^{\zeta(t)} \cdot X_{21} \tag{7.18}$$

Finally ΔX_2 can be computed using equation 7.7, therefore combining equation 7.5 and 7.1, the lateral forces can be obtained:

$$Y_1(t) = \frac{\dot{r}(t) \cdot J_{zz} + m(t) \cdot a_y(t) \cdot a2 - \Delta X_2(t) \cdot t_2}{l} \tag{7.19}$$

$$Y_2(t) = m(t) \cdot a_y(t) - Y_1(t) \tag{7.20}$$

7.2.3 Rear wheels with tires hypotesis

If a more accurate estimation of the lateral forces is required, the constitutive equations of the tires must be involved. The provided data are not sufficient to use the wheel with tire model, as it involves vertical lateral transfer, roll angles, and possibly the characterization of the suspension.

7.2.4 Implementation

The evaluation of the lateral forces, using the double track model with rigid rear wheels, can be done with a for cicle, where the equations of the section 7.2.2 are involved:

```matlab
% Evaluating Y1 and Y2 using double track model and rigid wheels
Y1_d = zeros(n,1);
Y2_d = zeros(n,1);
X21 = zeros(n,1);
X22 = zeros(n,1);
eta = 0.5;                  % Differential efficency
Rr = 0.36;                  % Rolling radius of the wheels
chi = 1000;                 % Time constant
t2 = carreggiata_rear(1);

for i = 1:n
    u21 = speed(i) - omega_z(i)*t2/2;
    u22 = speed(i) + omega_z(i)*t2/2;

    omega21 = u21/Rr;
    omega22 = u22/Rr;

    delta_omega = omega22 - omega21;

    zeta = atan(chi*delta_omega)/pi/2;

    X2 = massa_vettura(i)*ax(i) + Xa(i);

    X21(i) = X2 / ( eta^zeta +1 );
    X22(i) = X21(i) * eta^zeta;

    delta_X2 = X22(i) - X21(i);

    Y1_d(i) = ( d_omega_z_f(i)*Jzz(1) + massa_vettura(i)*ay(i)*a2 - delta_X2*t2 )
        / passo_vettura(1);
    Y2_d(i) = massa_vettura(i)*ay(i) - Y1_d(i);
end
```

7.2.5 Results

The results obtained with single and double track model are plotted together, to highlight the differences between the two models:

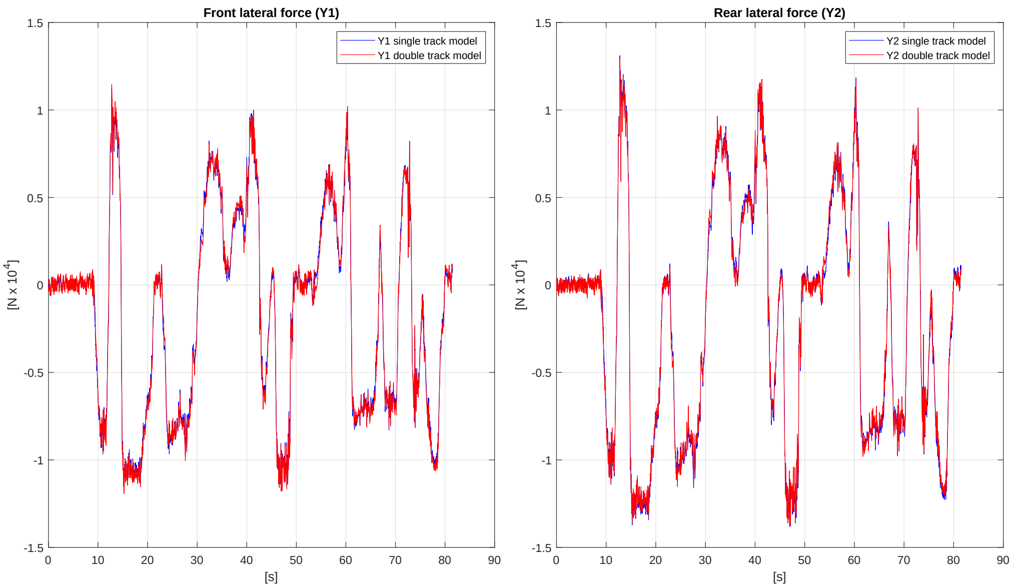

Figure 7.3: Front and rear lateral forces using single track model and double track model with rigid rear wheels

As can be seen in figure 11.1, there are no significant differences between the two models.

8 Yaw moment evaluation

The yaw moment is defined as:

$$N(t) = J_{zz}\dot{r}(t) \tag{8.1}$$

Therefore, the yaw moment can be compuyed directly from the telemetry data.

The implementation can be done with a for cicle:

```
% Evaluating yaw moment N
N = zeros(n,1);
for i = 1:n
    N(i) = d_omega_z_f(i)*Jzz(1);
end
```

The following yaw moment is obtained:

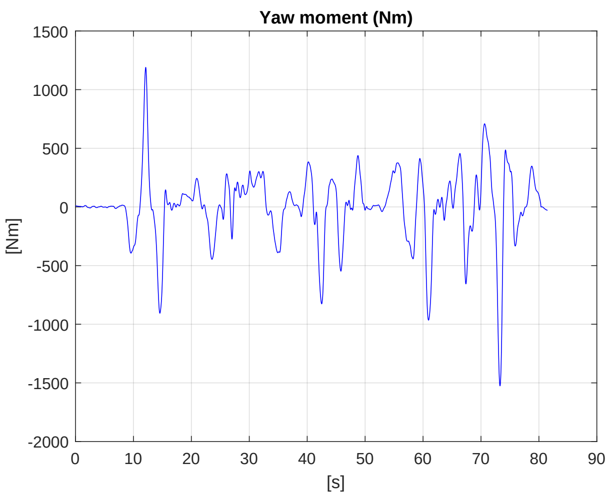

Figure 8.1: Yaw moment

9 g-g plot

The g-g plot is a diagram that have along the x axis the a_x and along the y axis the a_y, both of them expressed in g. It can be useful to visualize the dynamic limits of the vehicle, in particular the maximum value of a_x shows the limitation due to the power of the engine, the minimum value shows the braking system and grip limits; also the maximum and minimum values of a_y shows the grip limitations of the vehicle.

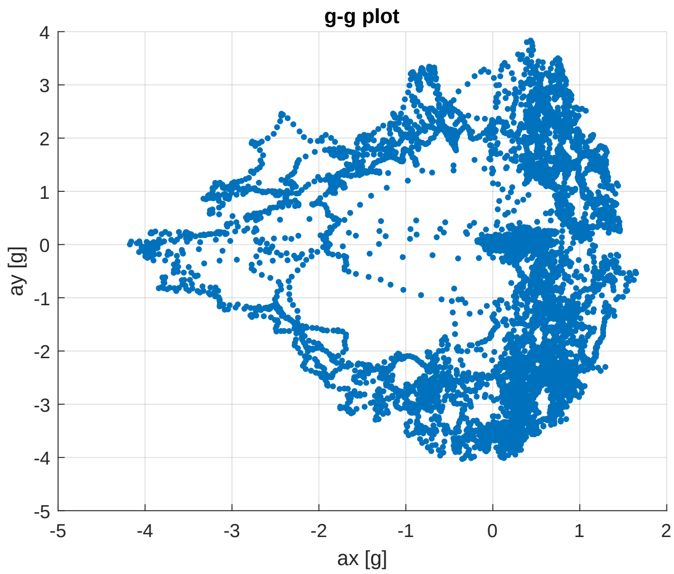

Figure 9.1: g-g plot

10 Fixed centrodes

The fixed centrode for a generic curve is described by the parametric equations $(x_f(\hat{t}), y_f(\hat{t}))$, expressed in the ground-fixed frame S_0 defined in the section 4.1, the following equations are required for its computation:

$$R(\hat{t}) = \frac{u(\hat{t})}{r(\hat{t})} \tag{10.1}$$

$$S(\hat{t}) = -\frac{v(\hat{t})}{r(\hat{t})} \tag{10.2}$$

$$x_f(\hat{t}) = x_0^G(\hat{t}) - S(\hat{t})cos\psi(\hat{t}) + R(\hat{t})sin\psi(\hat{t}) \tag{10.3}$$

$$y_f(\hat{t}) = y_0^G(\hat{t}) - S(\hat{t})sin\psi(\hat{t}) - R(\hat{t})cos\psi(\hat{t}) \tag{10.4}$$

Where the parameter $\hat{t}$ is varied from the start time to the end time of the curve.

10.1 Implementation

For the implementation, it is necessary to proceed with several steps:

1. Filtering the data required for the computation of the parametric equations 10.3 10.4, this operation is necessary to make the fixed centrodes more smooth.
2. Cumputing R and S for each time sample;
3. Defining a threshold to distinguish the curves, whose fixed centroids will be calculated, from the straight lines;
4. Calculate the fixed centrodes of the curves identified in the previous point;
5. Finding the starting sample time and ending sample time of all curves, in that way, the range of variation of the parameter $\hat{t}$ is known, and the fixed centrode of a specific curve can be plotted.

One of the possibles implementation is presented over here:

```
%% Fixed certrodes

% Filtering data
speed = smoothdata(speed, "loess","SmoothingFactor",0.25);
v = smoothdata(v, "loess","SmoothingFactor",0.25);
c_yaw = smoothdata(c_yaw, "loess","SmoothingFactor",0.25);
s_yaw = smoothdata(s_yaw, "loess","SmoothingFactor",0.25);
yaw = atan2(s_yaw, c_yaw);

% Computing R and S for each time sample
R = speed./omega_z_f;
S = -v./omega_z_f;
xc = zeros(n,1);
yc = zeros(n,1);

% Defining the threshold to distinguish the curves
radius_threshold = 450;

% Finding fixed centrodes of the curves
for i=1:n
    if sqrt( R(i)^2 + S(i)^2 ) < radius_threshold
        GC_v = [S(i); R(i)];
        GC_f = [-cos( yaw(i) ) sin( yaw(i) ); -sin( yaw(i) ) -cos( yaw(i) )]*GC_v;
        OC_f = GC_f + [xg(i); yg(i)];
        xc(i) = OC_f(1);
        yc(i) = OC_f(2);
    end
end

% Finding starting sample time and ending time of all curves
n_s = 0;                  % Vector of the starting times of curves
n_e = 0;                  % Vector of the ending times of curves
j = 1;
k = 1;

if xc(1) ~= 0
    n_s(j) = 1;
    j = j + 1;
else
    for i=2:n
        if (xc(i) ~= 0) && (xc(i-1) == 0)
            n_s(j) = i;
            j = j+1;
        elseif (xc(i) == 0) && (xc(i-1) ~= 0)
                n_e(k) = i-1;
                k = k+1;
        end
    end
end
```

10.2 Results

The fixed centrodes and the relative curves are plotted together:

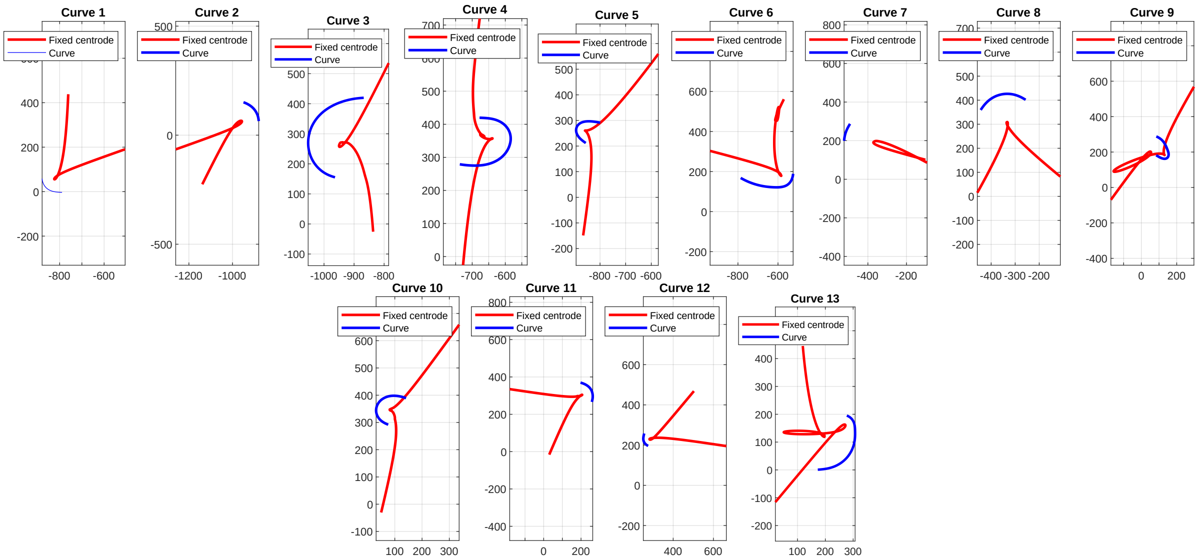

Figure 10.1: Fixed centrodes and their curves

11 Moving centrode of one curve

The moving centrode for a generic curve is described by the parametric equations $(x_m^f(t,\hat{t}), y_m^f(t,\hat{t}))$, expressed in the ground-fixed frame S_0 defined in the section 4.1, the following equations are required for its computation:

$$x_m^f(t,\hat{t}) = x_0^G(t) - S(\hat{t})cos\psi(t) + R(\hat{t})sin\psi(t) \tag{11.1}$$
$$y_m^f(t,\hat{t}) = y_0^G(t) - S(\hat{t})sin\psi(t) - R(\hat{t})cos\psi(t) \tag{11.2}$$

The parameter to draw the moving centrode is $\hat{t}$, while t sets the instant of time.

The goal of this section is to draw the moving centrode of the curve eight, for the starting, the intermediate and final sample times of that curve.

11.1 Implementation

One of the possible implementations is presented over here:

```
% Moving centrode
n_c8 = (n_s_c8:1:n_e_c8)';                % Samples of curve8
n1= n_c8(1);                              % Starting sample
n2 = n_c8(floor(length(n_c8)/2));         % Intermediate sample
n3 = n_c8(end);                           % Final sample
n_c = [n1;n2;n3];

xm = zeros( length(n_c8), 3);
ym = zeros( length(n_c8), 3);

for j=1:3
    for i=1:length(n_c8)
        xm(i,j) = xg(n_c(j)) - S( i + n1 - 1 )*cos( yaw( n_c(j) ) ) + R( i + n1 -
            1 )*sin( yaw( n_c(j) ) );
        ym(i,j) = yg(n_c(j)) - S( i + n1 - 1 )*sin( yaw( n_c(j) ) ) - R( i + n1 -
            1 )*cos( yaw( n_c(j) ) );
    end
end
```

11.2 Results

The moving centrodes at the three instant, the fixed centrode and the curve8 are plotted together:

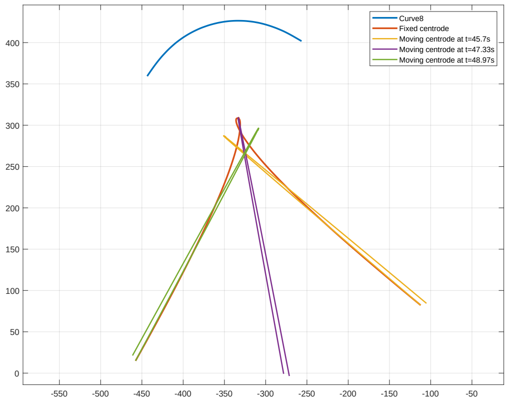

Figure 11.1: Moving centrodes at the three instant, fixed centrodes and curve8

12 Power limited curves and grip limited curves

Power limited curves and grip limited curves are supposed to be respectively, the section of the curves where the pilot is pressing the gas pedal to the maximum and the section of the curves where the pilot is pressing the brakes pedal to the maximum. Therefore, to plot power and grip limited curves, it is necessary to verify the following conditions in the time sample relative to the curves:

- If the opening of the valves of the motor engine ($farf$) is over their 89% at a certain time of the curve, at that time the coordinates of the center of mass of the vehicle will be part of the power limited curves;
- If the pressure of the oil in the brake system (p_brake) is over a threshold value corresponding to $1.5Pa$ at a certain time of the curve, at that time the coordinates of the center of mass of the vehicle will be part of the grip limited curves.

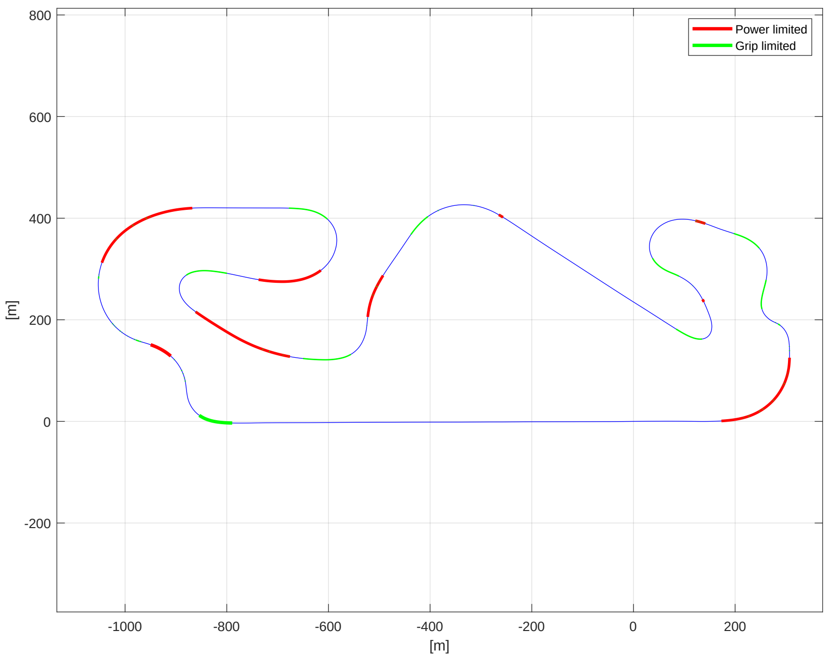

Figure 12.1: g-g plot