Universitat Politècnica de Catalunya

Master Thesis

Appendixes

Study of stability control systems applied to a racing car

Author:
Albert Inglés Navarrete

Supervisor:
Dr. David Gonzàlez

A thesis submitted in fulfillment of the requirements for the degree of Màster Universitari en Enginyeria Aeronàutica (MUEA)

in the

Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

January 13th, 2021

Declaration of Authorship

I, Albert Inglés Navarrete, declare that this thesis titled, 'Study of stability control systems applied to a racing car' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: ______________________________

Date: ______________________________

Contents

List of Figures

Appendix A

Tire Model: Magic Formula 6.1

In this appendix, the equations and description of the Magic Formula 6.1 tire model will be performed.

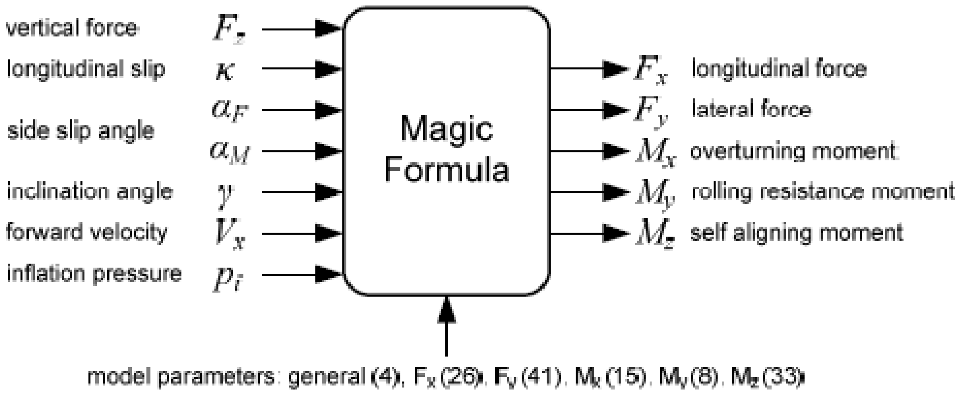

Figure A.1: Inputs and outputs of the Magic Formula. Source: [1], pag. 11

The first step is to dimensionless the vertical force and the pressure:

$$df_z = \frac{F_z - F_{z,0}}{F_{z,0}} \tag{A.1}$$

$$dP_i = \frac{P_i - P_{i,0}}{P_{i,0}} \tag{A.2}$$

A.1 Longitudinal force F_x

$$F_x = (D_x \sin[C_x \arctan\{B_x\kappa_x - E_x \arctan(B_x\kappa_x - \arctan(B_x\kappa_x))\}] + S_{V_x}) \cdot G_{x\alpha} \quad \text{(A.3)}$$

Pure slip:

$$\kappa_x = \kappa + S_{Hx} \quad \text{(A.4)}$$

$$C_x = P_{Cx1}\lambda_{Cx} \quad \text{(A.5)}$$

$$D_x = \mu_x F_z \quad \text{(A.6)}$$

$$\mu_x = (p_{Dx1} + p_{Dx2}df_z)(1 - p_{Dx3}\gamma^2)(1 + p_{px3}dp_i + p_{px4}dp_i^2)\lambda_{\mu x} \quad \text{(A.7)}$$

$$E_x = (p_{Ex1} + p_{Ex2}df_z + p_{Ex3}df_z^2)(1 - p_{Ex4}\mathrm{sgn}(\kappa_x))\lambda_{Ex} \quad \text{(A.8)}$$

$$K_{x\kappa} = (p_{Kx1} + p_{Kx2}df_z)\exp(p_{Kx3}df_z)(1 + p_{px1}dp_i + p_{px2}dp_i^2)F_z\lambda_{Kx\kappa} \quad \text{(A.9)}$$

$$B_x = \frac{K_{x\kappa}}{C_x D_x} \quad \text{(A.10)}$$

$$S_{Hx} = (p_{Hx1} + p_{Hx2}df_z)\lambda_{Hx} \quad \text{(A.11)}$$

$$S_{Vx} = (p_{Vx1} + p_{Vx2}df_z)F_z\lambda_{Vx}\lambda_{\mu x} \quad \text{(A.12)}$$

Combined longitudinal correction factor:

$$G_{x\alpha} = \frac{\cos[C_{x\alpha} \arctan\{B_{x\alpha}\alpha_s - E_{x_\alpha}(B_{x\alpha}\alpha_s - \arctan(B_{x\alpha}\alpha_s))\}]}{\cos[C_{x\alpha} \arctan\{B_{x\alpha}S_{Hx\alpha} - E_{x_\alpha}(B_{x\alpha}S_{Hx\alpha} - \arctan(B_{x\alpha}S_{Hx\alpha}))\}]} \quad \text{(A.13)}$$

$$\alpha_s = \alpha_F + S_{Hx\alpha} \quad \text{(A.14)}$$

$$B_{x\alpha} = (r_{Bx1} + r_{Bx3}\gamma^2)\cos(\arctan(r_{Bx2}\kappa))\lambda_{x\alpha} \quad \text{(A.15)}$$

$$C_{x\alpha} = r_{Cx1} \quad \text{(A.16)}$$

$$C_{x\alpha} = r_{Ex1} + r_{Ex2}df_z \quad \text{(A.17)}$$

$$S_{Hx\alpha} = r_{Hx1} \quad \text{(A.18)}$$

A.2 Overturning moment M_x

$$M_x = R_0 F_z \lambda_{Mx}\{q_{sx1}\lambda_{vMx} - q_{sx2}\gamma(1 + P_{pMx1}dp_i) - q_{sx12}\gamma \mid \gamma \mid + q_{sx3}\frac{F_y}{F_{z0}} \\ + q_{sx4}\cos\left[q_{sx4}\arctan\left(\left(q_{sx6}\frac{F_z}{F_{z0}}\right)^2\right)\right]\cdot\sin\left[q_{sx7}\gamma + q_{sx8}\arctan\left(\left(q_{sx9}\frac{F_y}{F_{z0}}\right)^2\right)\right] \\ + q_{sx10}\arctan\left(q_{sx11}\frac{F_z}{F_{z0}}\right)\gamma\} + R_0 F_y \lambda_{Mx}\{q_{sx13} + q_{sx13} \mid \gamma \mid\} \quad \text{(A.19)}$$

A.3 Rolling resistance moment M_y

$$M_y = -R_0 F_{z0}\lambda_{My}(q_{sy1} + q_{sy2}\frac{F_x}{F_{z0}} + q_{sy3}\left|\frac{V_x}{V_{ref}}\right| + q_{sy4}\left(\frac{V_x}{V_{ref}}\right)^4 + \\ q_{sy5}\gamma^2 + q_{sy6}\frac{F_z}{F_{z0}}\gamma^2)\left(\frac{F_z}{F_{z0}}\right)^{q_{sy7}}\left(\frac{p}{p_0}\right)^{q_{sy8}} \quad \text{(A.20)}$$

A.4 Lateral force F_y

$$F_y = G_{y\kappa} F_{yp} + S_{V_{y\kappa}} \tag{A.21}$$

Pure slip:

$$F_{yp} = D_y \sin\left[C_y \arctan\{B_y\alpha_y - E_y(B_y\alpha_y - \arctan(B_y\alpha_y))\}\right] + S_{V_y} \tag{A.22}$$

$$\alpha_y = \alpha + S_{Hy} \tag{A.23}$$

$$C_y = P_{Cy1}\lambda_{Cy} \tag{A.24}$$

$$D_y = \mu_y F_z \tag{A.25}$$

$$\mu_y = (p_{Dy1} + p_{Dy2}df_z)(1 - p_{Dy3}\gamma^2)(1 + p_{py3}dp_i + p_{py4}dp_i^2)\lambda_{\mu y} \tag{A.26}$$

$$E_y = (p_{Ey1} + p_{Ey2}df_z)(1 + p_{Ey5}\gamma^2 - (p_{Ey3} + p_{Ey4}\gamma)\text{sgn}(\alpha_y))\lambda_{Ey} \tag{A.27}$$

$$K_{y\alpha} = (p_{Ky1}F_{z0}(1 + p_{py1}dp_i)\sin\left[p_{Ky4}\arctan\left\{\frac{F_z}{(p_{Ky2} + p_{Ky5}\gamma^2)(1 + p_{py2}dp_i)F_{z0}}\right\}\right] (1 - p_{Ky3}|\gamma|)\lambda_{Ky\alpha} \tag{A.28}$$

$$K_{y\gamma} = (p_{Ky6} + p_{Ky7}df_z)(1 + p_{py5}dp_i)F_z\lambda_{Ky\gamma} \tag{A.29}$$

$$B_y = \frac{K_{y\alpha}}{C_y D_y} \tag{A.30}$$

$$S_{Hy} = S_{Hy0+S_{Hy\gamma}} \tag{A.31}$$

$$S_{Hy0} = (p_{Hy1} + p_{Hy2}df_z)\lambda_{Hy} \tag{A.32}$$

$$S_{Hy\gamma} = \frac{K_{y\gamma}\gamma - S_{Vy\gamma}}{K_{y\alpha}} \tag{A.33}$$

$$S_{Vy} = S_{Vy0+S_{Vy\gamma}} \tag{A.34}$$

$$S_{Vy0} = (p_{Vy1} + p_{Vy2}df_z)F_z\lambda_{Vy}\lambda_{\mu y} \tag{A.35}$$

$$S_{Vx} = (p_{Vy3} + p_{Vy4}df_z)F_z\gamma\lambda_{Ky\gamma}\lambda_{\mu y} \tag{A.36}$$

Combined lateral correction factor:

$$S_{Vy\kappa} = D_{Vy\kappa}\sin(r_{Vy5}\arctan(r_{Vy6}\kappa))\lambda_{Vy\kappa} \tag{A.37}$$

$$D_{Vy\kappa} = \mu_y F_z(r_{Vy1} + r_{Vy2}df_z + r_{Vy3}\gamma)\cos(\arctan(r_{Vy4}\alpha)) \tag{A.38}$$

$$G_{y\kappa} = \frac{\cos[C_{y\kappa}\arctan\{B_{y\kappa}\kappa_s - E_{y_\kappa}(B_{y\kappa}\kappa_s - \arctan(B_{y\kappa}\kappa_s))\}]}{\cos[C_{y\kappa}\arctan\{B_{y\kappa}S_{Hy\kappa} - E_{y_\kappa}(B_{y\kappa}S_{Hy\kappa} - \arctan(B_{y\kappa}S_{Hy\kappa}))\}]} \tag{A.39}$$

$$\kappa_s = \kappa + S_{Hy\kappa} \tag{A.40}$$

$$B_{y\kappa} = (r_{By1} + r_{By4}\gamma^2)\cos(\arctan(r_{By2}(\alpha - r_{By3})))\lambda_{y\kappa} \tag{A.41}$$

$$C_{y\kappa} = r_{Cy1} \tag{A.42}$$

$$C_{y\kappa} = r_{Ey1} + r_{Ey2}df_z \tag{A.43}$$

$$S_{Hy\kappa} = r_{Hy1} + r_{Hy2}df_z \tag{A.44}$$

A.5 Self-aligning moment M_z

$$M_z = -t\cdot F_{yp0}\cdot G_{y\kappa 0} + M_{zr} + s\cdot F_x \tag{A.45}$$

Where $F_{yp0}\cdot G_{y\kappa 0}$ is the combined slip side force with zero inclination angles $\gamma = 0$.

$$\alpha_t = \alpha + S_{Ht} \tag{A.46}$$

$$S_{Ht} = q_{Hz1} + q_{Hz2}df_z + (q_{Hz3} + q_{Hz4}df_z)\gamma \tag{A.47}$$

$$\alpha_r = \alpha + S_{Hy} + \frac{S_{Vy}}{K_{y\alpha}} \tag{A.48}$$

$$\alpha_{t,eq} = \arctan(\sqrt{\tan^2(\alpha_t) + \left(\frac{K_{x\kappa}}{K_{y\alpha}}\right)^2\kappa^2}\mathrm{sgn}(\alpha_t)) \tag{A.49}$$

$$\alpha_{r,eq} = \arctan(\sqrt{\tan^2(\alpha_r) + \left(\frac{K_{x\kappa}}{K_{y\alpha}}\right)^2\kappa^2}\mathrm{sgn}(\alpha_r)) \tag{A.50}$$

$$s = \left(s_{sz1} + s_{sz1}\left(\frac{F_y}{f_{z0}}\right) + (s_{sz1} + s_{sz1}df_z)\gamma\right)R_0\lambda_s \tag{A.51}$$

Pneumatic trail t:

$$t = D_t\cos\left[C_t\arctan\{B_t\alpha_{t,eq} - E_t(B_t\alpha_{t,eq} - \arctan(B_t\alpha_{t,eq})\}\right]\cos(\alpha) \tag{A.52}$$

$$B_t = (q_{Bz1} + q_{Bz2}df_z + q_{Bz3}df_z^2)(1 + q_{Bz4}q_{Bz5}|\gamma|)\frac{\lambda_{Ky\alpha}}{\lambda_{\mu_y}} \tag{A.53}$$

$$C_t = q_{Cz1} \tag{A.54}$$

$$D_t = (q_{Dz1} + q_{Dz2} df_z)(1 - q_{Dz1} dp_i)(1 + q_{Dz3}\gamma + q_{Bz4}\gamma^2) F_z \frac{R_0}{F_{z0}} \lambda_t \tag{A.55}$$

$$E_t = (q_{Ez1} + q_{Ez2} df_z + q_{Ez3} df_z^2) \left(1 + (q_{Ez4} q_{Bz5}\gamma) \left(\frac{2}{\pi}\right) \arctan(B_t C_t \alpha_t)\right) \tag{A.56}$$

Residual moment M_{zr}:

$$M_{zr} = D_r \cos\left[\arctan\{B_r \alpha_{r,eq}\right] \cos(\alpha) \tag{A.57}$$

$$B_r = q_{Bz9} \frac{\lambda_{Ky\alpha}}{\lambda_{\mu y}} + q_{Bz10} B_y C_y \tag{A.58}$$

$$\begin{aligned} D_r = [(q_{Dz6} + (q_{Dz7} df_z)\lambda_r + ((q_{Dz8} + (q_{Dz9} df_z)(1 - p_{pz2} dp_i)\gamma \lambda_{Kz\gamma} \\ + (q_{Dz10} + (q_{Dz11} df_z)\gamma |\gamma| \lambda_{Kz\gamma}] F_z R_0 \lambda_{\mu y} \end{aligned} \tag{A.59}$$

A.6 Tire model Matlab code

```
function [Fx,Fy,Mx,My,Mz,cx,cy,cz,Rl]=Evaluate_Tyre_Model(inputs,coefficients,
    scaling_factors,Tot)
%% Index
        %Adimensionalization
        %Fx calculation
        %Fy calculation
        %Mx calculation
        %My calculation
        %Mz calculation
        %Radius calculation (Cancelled)
        %Contact patch calculation (Cancelled)
        %Stiffness calculation
%% Bibliography
    % Using UPC ecoRacing Pacejka Model extractet from the paper:
        % [4] An improved Magic Formula/Swift tyre model that can handle in ation
     pressure changes.
        % In Vehicle System Dynamics, volume 48, pages 337-352, 2010.
    %% Adimensionalization
    dFz=(inputs(1)-coefficients(1))/coefficients(1);
    dPi=(inputs(6)-coefficients(4))/coefficients(4);

    %% Fx calculation
    %Pure slip (34-42)
    Kxk=(coefficients(40)+coefficients(41)*dFz)*exp(coefficients(42)*dFz)*(1+
    coefficients(47)*dPi+coefficients(48)*dPi^2)*inputs(1)*scaling_factors(5);
    SHx=(coefficients(43)+coefficients(44)*dFz)*scaling_factors(11);
    SVx=(coefficients(45)+coefficients(46)*dFz)*inputs(1)*scaling_factors(2)*
    scaling_factors(13);
    kx=inputs(2)+SHx;
    Mux=(coefficients(33)+coefficients(34)*dFz)*(1-coefficients(35)*inputs(4)^2)
    *(1+coefficients(49)*dPi+coefficients(50)*dPi^2)*scaling_factors(2);
    Cx=coefficients(32)*scaling_factors(7);
    Dx=Mux*inputs(1);
    Ex=(coefficients(36)+coefficients(37)*dFz+coefficients(38)*dFz^2)*(1-
    coefficients(39)*sign(inputs(2)))*scaling_factors(9);
    Bx=Kxk/Cx/Dx;

    if Ex>1
        Error=['Error in the value of E: Ex=' Ex]; %#ok<NASGU>
        disp('Error')
        display(inputs)
        clear Error
    end

    %Combined slip (43-48)
    SHxa=coefficients(100);
    SAs=inputs(3)+SHxa;
    Bxa=(coefficients(96)+coefficients(98)*inputs(4)^2)*cos(atan(coefficients(97)
    *inputs(2)))*scaling_factors(19);
    Cxa=coefficients(99);
    Exa=coefficients(101)+coefficients(102)*dFz;

```

```matlab
Gxa=cos(Cxa*atan(Bxa*SAs-Exa*(Bxa*SAs-atan(Bxa*SAs))))/cos(Cxa*atan(Bxa*SHxa-Exa*(Bxa*SHxa-atan(Bxa*SHxa))));

if Exa>1
    Error=['Error in the value of E: Exa=' Exa]; %#ok<NASGU>
    disp('Error')
    display(inputs)
    clear Error
end

%Pacejka formula (33)
Fx=(Dx*sin(Cx*atan(Bx*kx-Ex*(Bx*kx-atan(Bx*kx))))+SVx)*Gxa;
%% Fy calculation
%Pure slip (53-66)
Kya=coefficients(14)*coefficients(1)*(1+coefficients(27)*dPi)*sin(coefficients(17)*atan(inputs(1)/...
    ((coefficients(15)+coefficients(18)*inputs(4)^2)*(1+coefficients(28)*dPi)*coefficients(1))))*(1-coefficients(16)*abs(inputs(4)))*scaling_factors(6);
Kyia=(coefficients(19)+coefficients(20)*dFz)*(1+coefficients(31)*dPi)*inputs(1)*scaling_factors(15);
SVy0=inputs(1)*(coefficients(23)+coefficients(24)*dFz)*scaling_factors(3)*scaling_factors(14);
SVyia=inputs(1)*(coefficients(25)+coefficients(26)*dFz)*inputs(4)*scaling_factors(3)*scaling_factors(15);
SHy0=(coefficients(21)+coefficients(22)*dFz)*scaling_factors(12);
SHyia=(Kyia*inputs(4)-SVyia)/Kya;
SHy=SHy0+SHyia;
SVy=SVy0+SVyia;
SAy=inputs(3)+SHy;
Muy=(coefficients(6)+coefficients(7)*dFz)*(1-coefficients(8)*inputs(4)^2)*(1+coefficients(29)*dPi+coefficients(30)*dPi^2)*scaling_factors(3);
Cy=coefficients(5)*scaling_factors(8);
Dy=Muy*inputs(1);
Ey=(coefficients(9)+coefficients(10)*dFz)*(1+coefficients(13)*inputs(4)^2-(coefficients(11)+coefficients(12)*inputs(4))*sign(SAy))*scaling_factors(10);
By=Kya/Cy/Dy;

if Ey>1
    Error=['Error in the value of E: Ey=' Ey]; %#ok<NASGU>
    disp('Error')
    display(inputs)
    clear Error
end

%Combined slip (67-74)
SHyk=coefficients(88)+coefficients(89)*dFz;
ks=inputs(2)+SHyk;
Byk=(coefficients(81)+coefficients(84)*inputs(4)^2)*cos(atan(coefficients(82)*(inputs(3)-coefficients(83))))*scaling_factors(20);
Cyk=coefficients(85);
Eyk=coefficients(86)+coefficients(87)*dFz;
DVyk=Muy*inputs(1)*(coefficients(90)+coefficients(91)*dFz+coefficients(92)*inputs(4))*cos(atan(coefficients(93)*inputs(3)));
SVyk=DVyk*sin(coefficients(94)*atan(coefficients(95)*inputs(2)))*scaling_factors(21);
```

```matlab

Gyk=cos(Cyk*atan(Byk*ks-Eyk*(Byk*ks-atan(Byk*ks))))/cos(Cyk*atan(Byk*SHyk-Eyk*(Byk*SHyk-atan(Byk*SHyk))));

if Eyk>1
    Error=['Error in the value of E: Eyk=' Eyk]; %#ok<NASGU>
    disp('Error')
    display(inputs)
    clear Error
end

%Pacejka formula (51-52)
Fy=Gyk*(Dy*sin(Cy*atan(By*SAy-Ey*(By*SAy-atan(By*SAy))))+SVy)+SVyk;

%% Mx calculation (49)
Mx=coefficients(2)*inputs(1)*scaling_factors(24)*(coefficients(107)*scaling_factors(26)-coefficients(108)*inputs(4)*(1+coefficients(121)*dPi)...
    -coefficients(118)*inputs(4)*abs(inputs(4))+coefficients(109)*Fy/coefficients(1)...
    +coefficients(110)*cos(coefficients(111)*atan((coefficients(112)*inputs(1)/coefficients(1))^2))*sin(coefficients(113)*inputs(4)...
    +coefficients(114)*atan(coefficients(115)*Fy/coefficients(1))+coefficients(116)*atan(coefficients(117)*inputs(1)/coefficients(1))*inputs(4))...
    +coefficients(2)*Fy*scaling_factors(24)*(coefficients(119)+coefficients(120)*abs(inputs(4)));

%% My calculation (50)
My=-coefficients(2)*coefficients(1)*scaling_factors(25)*(coefficients(122)+coefficients(123)*Fx/coefficients(1)...
     +coefficients(124)*abs(inputs(5)/coefficients(3))+coefficients(125)*(inputs(5)/coefficients(3))^4+coefficients(126)*inputs(4)^2 ...
     +coefficients(127)*inputs(1)/coefficients(1)*inputs(4)^2)*(inputs(1)/coefficients(1))^coefficients(128)*(inputs(6)/coefficients(4))^coefficients(129);

%% Mz calculation
%Combined slip force with zero inclination angle Fy@IA=0
if inputs(4)==0
    Fy0=Fy;
else
    %Pure slip (53-66)
    Kya=coefficients(14)*coefficients(1)*(1+coefficients(27)*dPi)*...
        sin(coefficients(17)*atan(inputs(1)/(coefficients(15)*(1+coefficients(28)*dPi)*coefficients(1))))*scaling_factors(6);
    SVy=inputs(1)*(coefficients(23)+coefficients(24)*dFz)*scaling_factors(3)*scaling_factors(14);
    SHy=(coefficients(21)+coefficients(22)*dFz)*scaling_factors(12);
    SAy=inputs(3)+SHy;
    Muy=(coefficients(6)+coefficients(7)*dFz)*(1+coefficients(29)*dPi+coefficients(30)*dPi^2)*scaling_factors(3);
    Cy=coefficients(5)*scaling_factors(8);
    Dy=Muy*inputs(1);
    Ey=(coefficients(9)+coefficients(10)*dFz)*(1-coefficients(11)*sign(SAy))*scaling_factors(10);
```

```matlab
        By=Kya/Cy/Dy;

        if Ey>1
            Error=['Error in the value of E: Ey=' Ey]; %#ok<NASGU>
            disp('Error')
            display(inputs)
            clear Error
        end

        %Combined slip (67-74)
        SHyk=coefficients(88)+coefficients(89)*dFz;
        ks=inputs(2)+SHyk;
        Byk=coefficients(81)*cos(atan(coefficients(82)*(inputs(3)-coefficients(83))))*scaling_factors(20);
        Cyk=coefficients(85);
        Eyk=coefficients(86)+coefficients(87)*dFz;
        DVyk=Muy*inputs(1)*(coefficients(90)+coefficients(91)*dFz)*cos(atan(coefficients(93)*inputs(3)));
        SVyk=DVyk*sin(coefficients(94)*atan(coefficients(95)*inputs(2)))*scaling_factors(21);

        Gyk=cos(Cyk*atan(Byk*ks-Eyk*(Byk*ks-atan(Byk*ks))))/cos(Cyk*atan(Byk*SHyk-Eyk*(Byk*SHyk-atan(Byk*SHyk))));

        if Eyk>1
            Error=['Error in the value of E: Eyk0=' Eyk]; %#ok<NASGU>
            disp('Error')
            display(inputs)
            clear Error
        end

        %Pacejka formula (51-52)
        Fy0=Gyk*(Dy*sin(Cy*atan(By*SAy-Ey*(By*SAy-atan(By*SAy))))+SVy)+SVyk;
    end

    %Equivalent slip (76-81)
    SHt=coefficients(75)+coefficients(76)*dFz+(coefficients(77)+coefficients(78)*dFz)*inputs(4);
    SAt=inputs(3)+SHt;
    SAr=inputs(3)+SHy+SVy/Kyia;
    SAteq=atan(sqrt((tan(SAt))^2+(Kxk/Kya)^2*inputs(2)^2))*sign(SAt);
    SAreq=atan(sqrt((tan(SAr))^2+(Kxk/Kya)^2*inputs(2)^2))*sign(SAr);

    %Pneumatic scrub (82)
    s=(coefficients(103)+coefficients(104)*(Fy/coefficients(1))+(coefficients(105)+coefficients(106)*dFz)*inputs(4))*coefficients(2)*scaling_factors(22);

    %Pneumatic trail (83-87)
    Bt=(coefficients(51)+coefficients(52)*dFz+coefficients(53)*dFz^2)*(1+coefficients(54)+coefficients(55)*abs(inputs(4)))*scaling_factors(15)/scaling_factors(3);
    Ct=coefficients(59);
    Dt=(coefficients(60)+coefficients(61)*dFz)*(1-coefficients(79)*dPi)*(1+coefficients(62)*inputs(4)+coefficients(63)*inputs(4)^2)*inputs(1)*coefficients(2)/...
```

```
        coefficients(1)*scaling_factors(17);
    Et=(coefficients(70)+coefficients(71)*dFz+coefficients(72)*dFz^2)*(1+(coefficients(73)+coefficients(74)*inputs(4))*2/pi*atan(Bt*Ct*SAt));
    t=Dt*cos(Ct*atan(Bt*SAteq-Et*(Bt*SAteq-atan(Bt*SAteq))))*cos(inputs(3));

    %Resudial moment (88-90)
    Br=coefficients(57)*scaling_factors(15)/scaling_factors(3)+coefficients(58)*By*Cy;
    Dr=((coefficients(64)+coefficients(65)*dFz)*scaling_factors(18)+(coefficients(66)+coefficients(67)*dFz)*(1-coefficients(80)*dPi)*inputs(4)...
        *scaling_factors(16)+(coefficients(68)+coefficients(69)*dFz)*inputs(4)*abs(inputs(4))*scaling_factors(16))*inputs(1)*coefficients(2)*scaling_factors(3);
    Mzr=Dr*cos(atan(Br*SAreq))*cos(inputs(3));

    %Pacejka formula (75)
    Mz=-t*Fy0+Mzr+s*Fx;
    cx=0;
    cy=0;
    cz=0;
    Rl=0;
    if Tot==1
        %% Radius calculation
        c=coefficients(1)/coefficients(2)*sqrt(coefficients(133)^2+4*coefficients(134))*(1+coefficients(138)*dPi); %Vertical stiffness [N/m] used in (5)
        w=(1+inputs(2))*inputs(5)/coefficients(2); %Initial point of angular speed [rad/s] (6) supposing Re=R0
        for i=1:3 %Resolving of Rfr, Re and w
            wo=w;
            Rfr=coefficients(2)*(coefficients(130)+coefficients(131)*(wo*coefficients(2)/coefficients(3))^2); %Free tyre radius [m] (1)
            Re=Rfr-coefficients(1)/c*(coefficients(141)*atan(coefficients(142)*inputs(1)/coefficients(1))+coefficients(143)*inputs(1)/coefficients(1));...
                %Effective radius [m] (7)
            w=(1+inputs(2))*inputs(5)/Re; %Angular speed [rad/s] (6)
            if abs(w-wo)<0.1
                break;
            end
        end
        if abs(w-wo)>0.1
            Error=['Error in the convergence of the agular velocity. Last iteration error of ',abs(w-wo),' rad/s.'];
            disp(Error);
            display(inputs);
            clear Error
        end

        k3=(1+coefficients(132)*coefficients(2)/coefficients(3)*abs(w)-(coefficients(136)*Fx/coefficients(1))^2-(coefficients(137)*Fy/coefficients(1))^2)...
          *(1+coefficients(138)*dPi)*coefficients(1); %Terms part of (3)

        deflection=(-coefficients(133)/coefficients(2)+sqrt(coefficients(133)^2/coefficients(2)^2+4*coefficients(134)/coefficients(2)^2*inputs(1)/k3))...
```

```
                    /(2*coefficients(134)/coefficients(2)^2); %Deflection [m]
    resulting of solve (3)

        Rl=min(Rfr,Rfr-deflection); %Loaded radius [m] (2)
%          %% Contact patch calculation
%          a=coefficients(2)*(coefficients(145)*inputs(1)/c/coefficients(2)+
    coefficients(144)*sqrt(inputs(1)/c/coefficients(2)));...
%            %Half of the contact patch length [m] (9)
%          b=coefficients(146)*(coefficients(148)*inputs(1)/c/coefficients(2)+
    coefficients(147)*(inputs(1)/c/coefficients(2))^(1/3));...
%            %Half of the contact patch width [m] (10)
        %% Stiffness calculation
        cx=coefficients(153)*(1+coefficients(154)*dFz+coefficients(155)*dFz^2)
    *(1+coefficients(156)*dPi);   %Longitudinal stiffness [N/m] (17)
        cy=coefficients(149)*(1+coefficients(150)*dFz+coefficients(151)*dFz^2)
    *(1+coefficients(152)*dPi);   %Lateral      stiffness [N/m] (18)
        cz=k3*(coefficients(133)/coefficients(2)+2*coefficients(134)/coefficients
    (2)^2*deflection);          %Vertical     stiffness [N/m] resulting of dFz/dRl
     (3)
    end
end
```

Appendix B

Simulink vehicle model

In this appendix, Simulink vehicle model will be shown.Starting from the general overview of the Simulink blocks and, finally, entering in detail in each function of the model.

B.1 Simulink blocks and connections

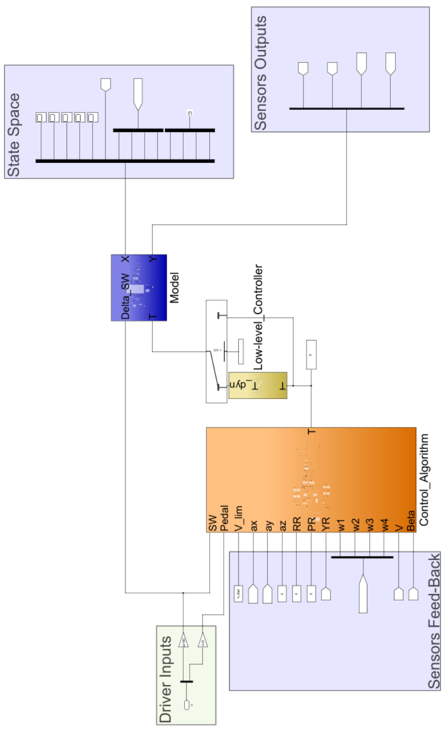

FIGURE B.1: Simulink model general overview.

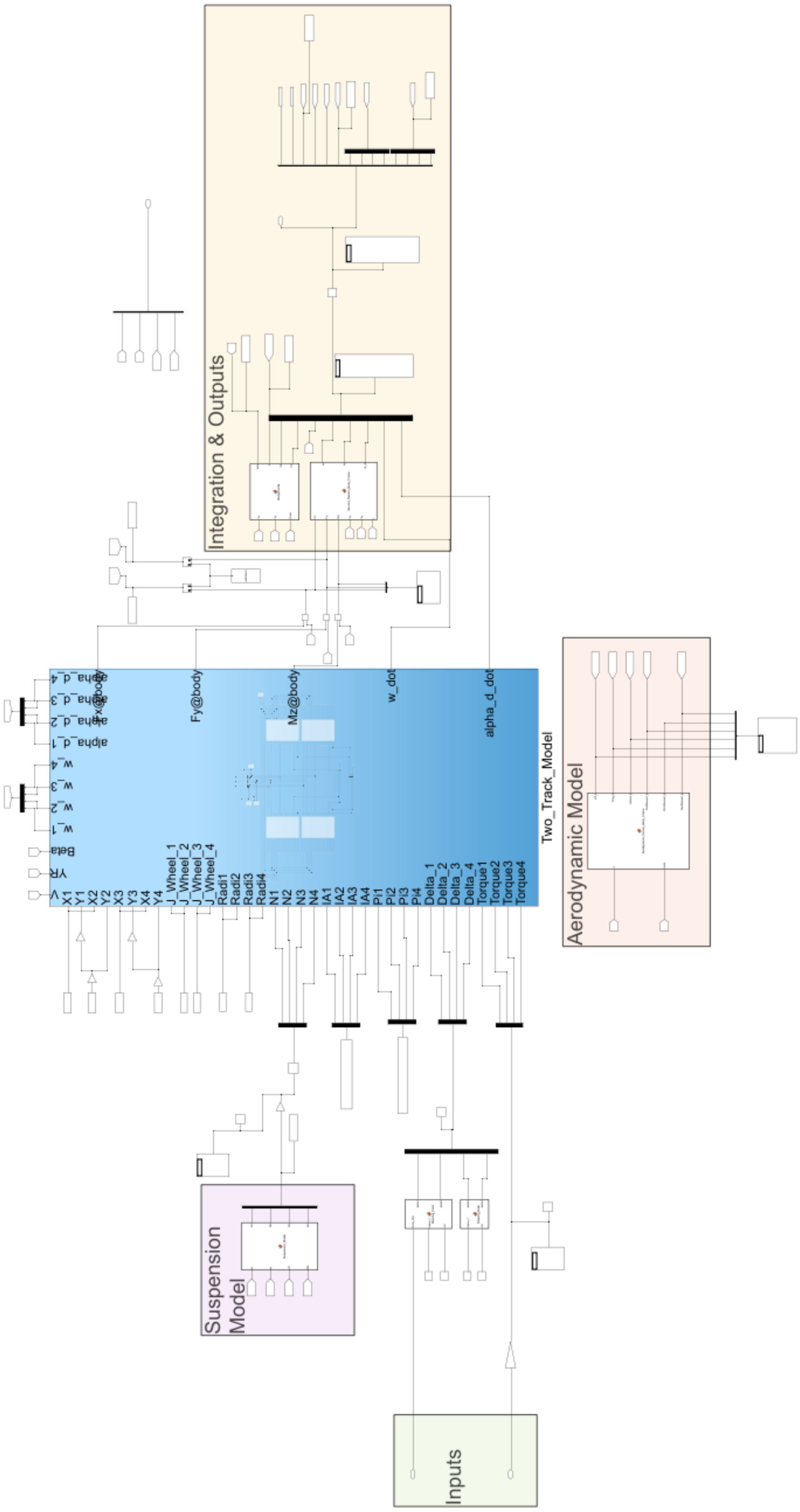

FIGURE B.2: Simulink vehicle physical model overview.

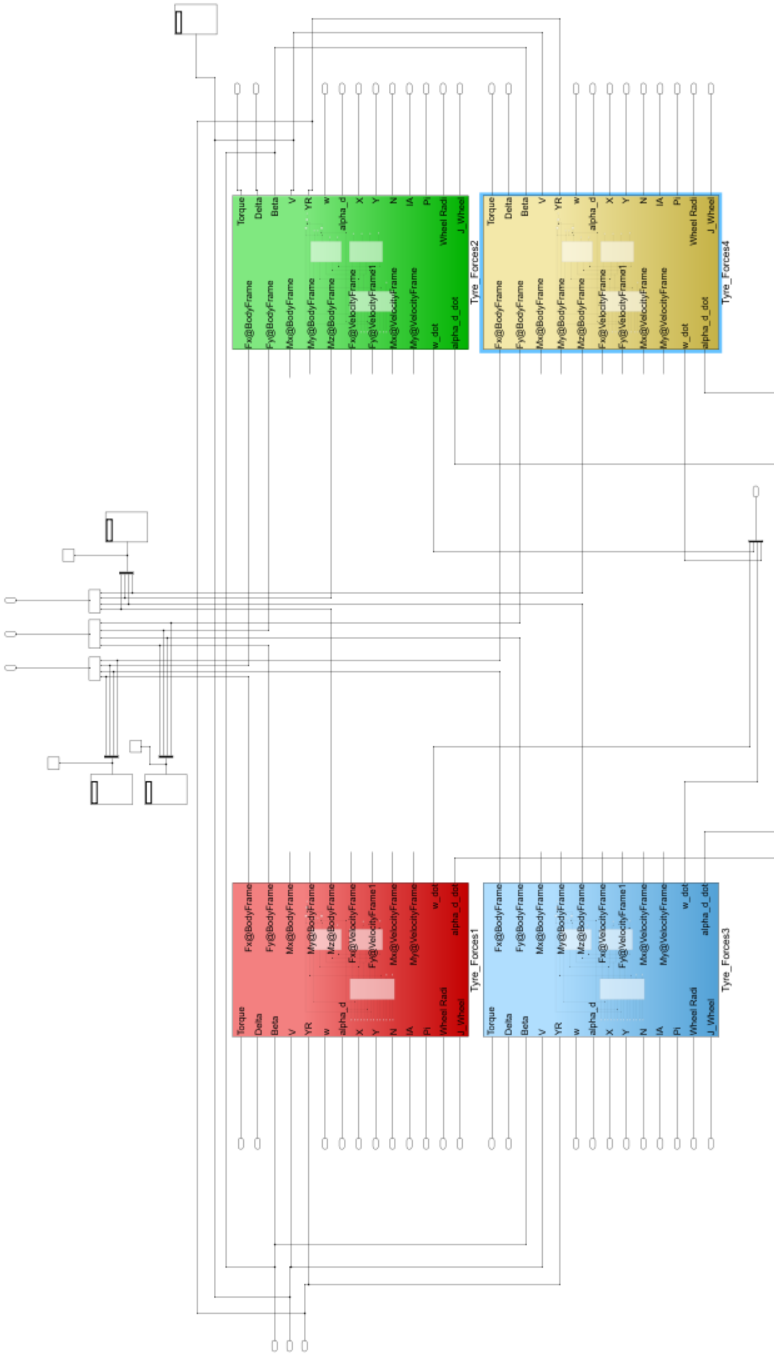

Figure B.3: Simulink two-track model overview.

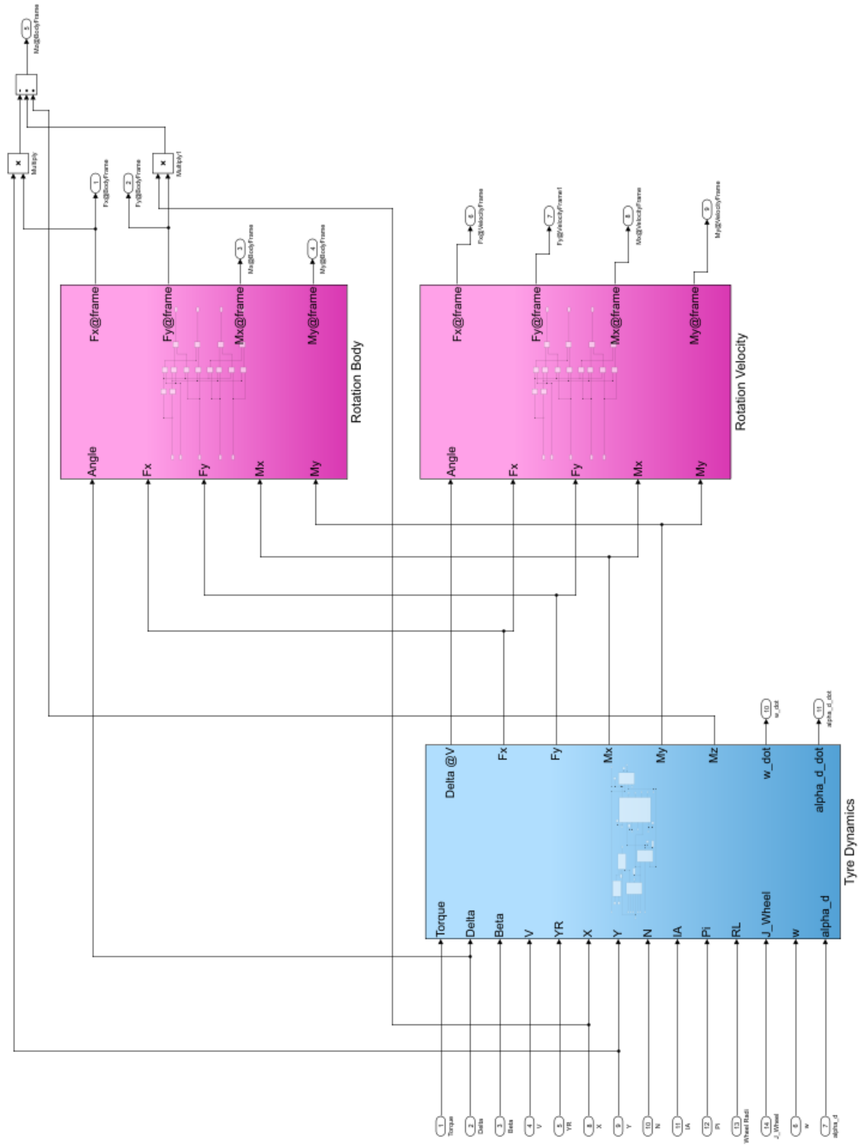

FIGURE B.4: Simulink tire dynamics and reference system projection.

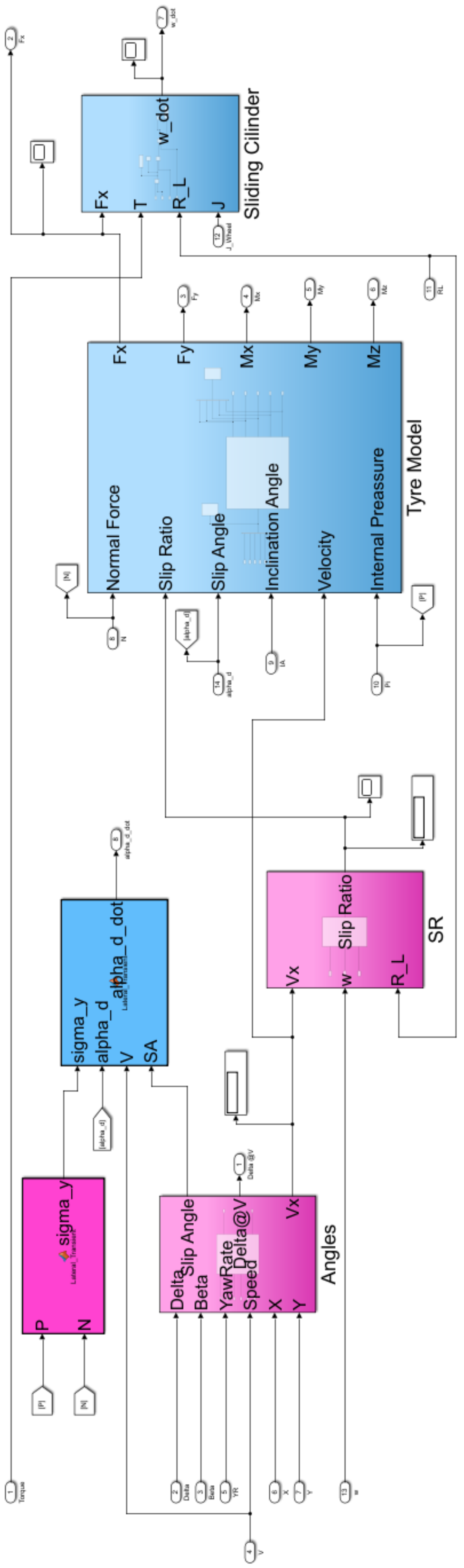

Figure B.5: Simulink tire dynamics and Magic Formula 6.1 tire model.

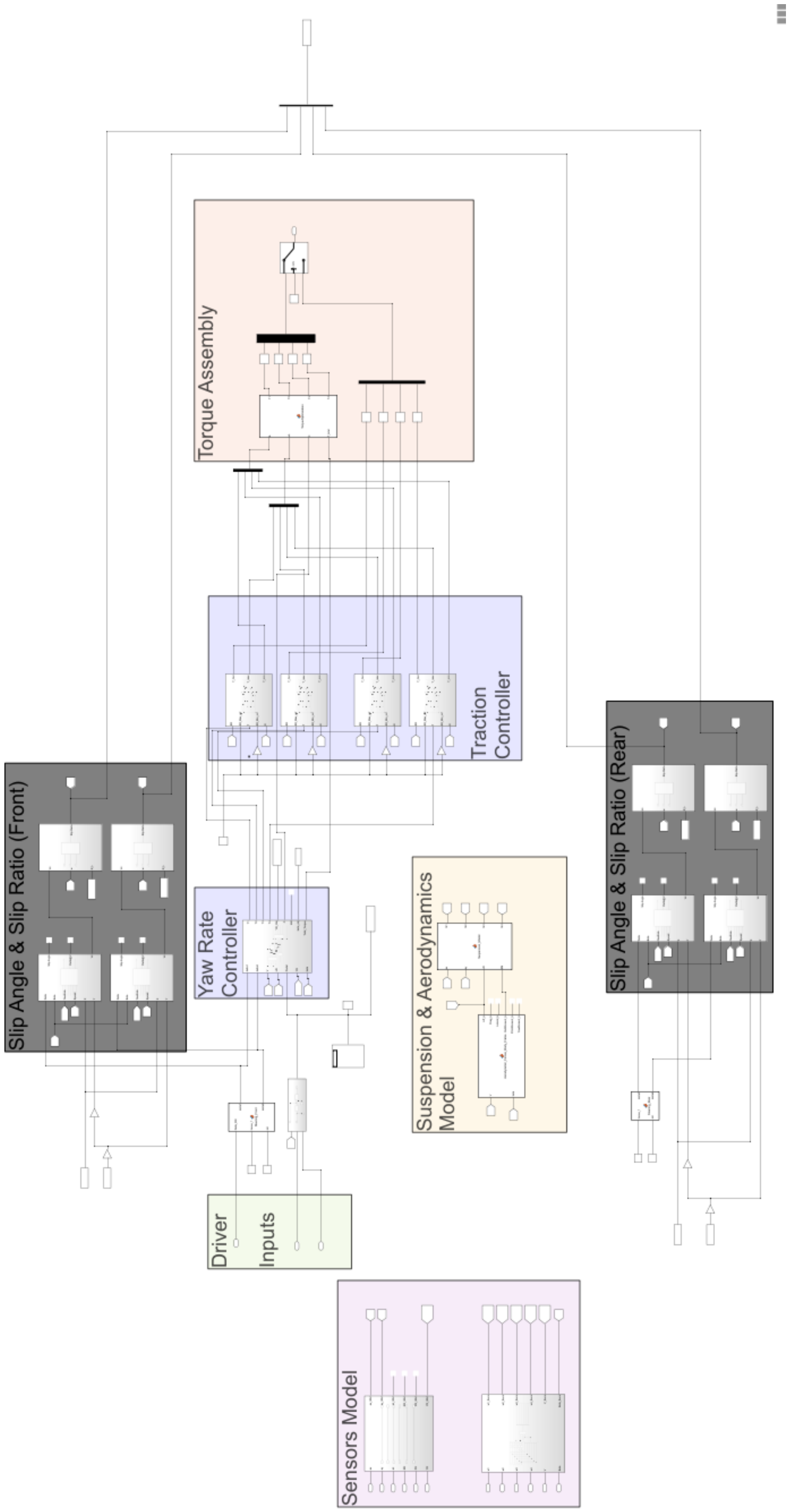

Figure B.6: Simulink control algorithm general overview.

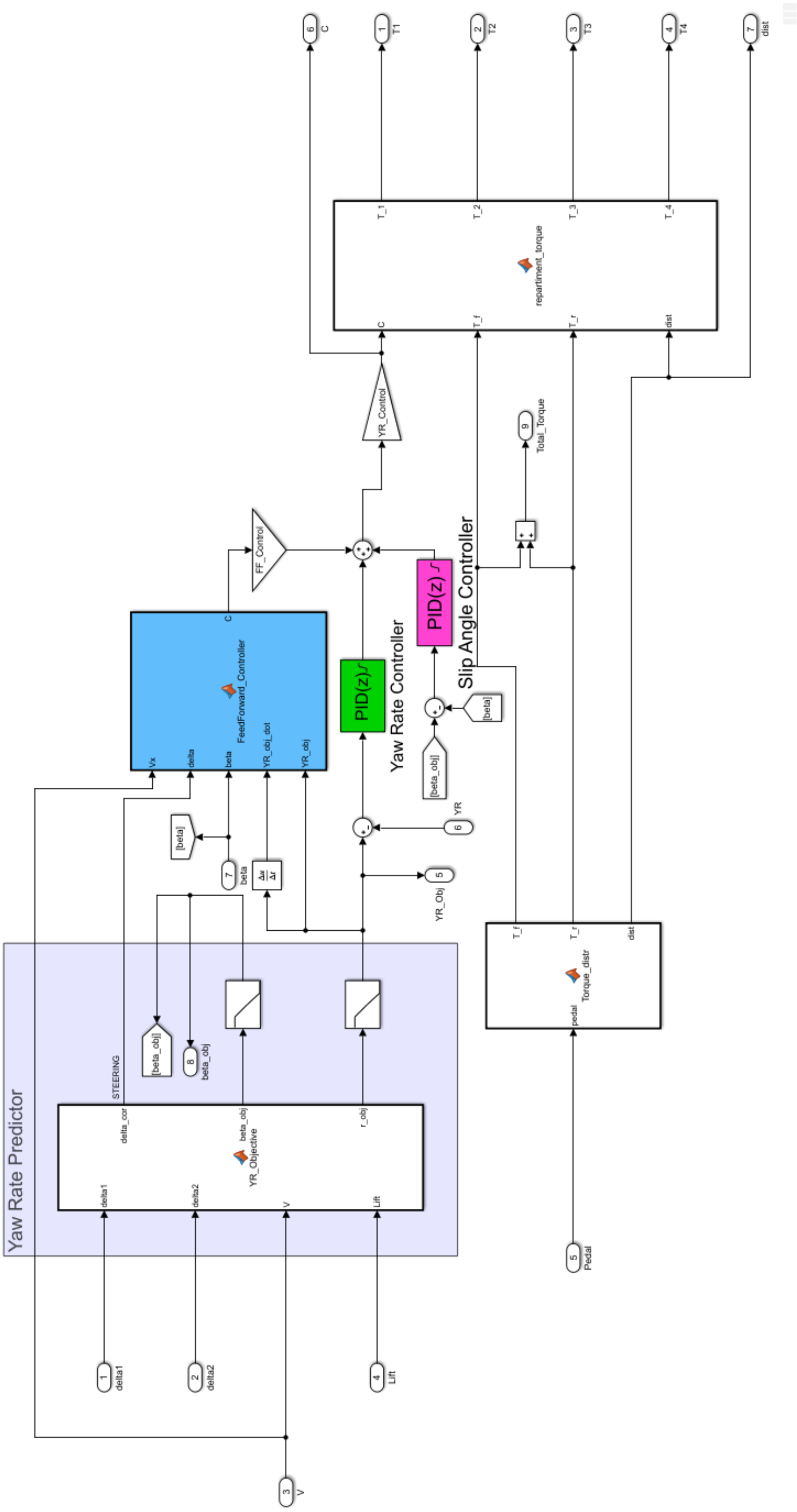

FIGURE B.7: Simulink yaw rate control algorithm general detailed.

B.2 Matlab and Simulink codes

FIGURE B.8: QR that give access to the Matlab and Simulink codes.

References

[1] I. J. M. Besselink, A. J. C. Schmeitz, and H. B. Pacejka. An improved Magic Formula/Swift tyre model that can handle inflation pressure changes. In *Vehicle System Dynamics*, volume 48, pages 337–352, 2010. ISBN 0042-3114. doi: 10.1080/00423111003748088.