ENPM667 Project 1

Holistic Adaptive Muli-Model Predictive Control

for the Path-following of 4WID Autonomous Vehicle

ADITYA VARADARAJ

(UID: 117054859)

SAURABH PRAKASH PALANDE

(UID: 118133959)

(SECTION: 0101)

M.Eng. Robotics (PMRO),

University of Maryland - College Park, College Park, MD - 20742, USA.

Date of Submission: 25th November, 2021

LIST OF FIGURES

LIST OF TABLES

CONTENTS

Abstract

The journal paper selected [1] proposes a control method for path-following of a 4-Wheel Independent Drive (4WID) Autonomous car. Firstly, it assumes a single track model for the vehicle and derive system model for the vehicle-path system. Using the system model, they find the continuous state-space representation of the system. Then, they discretize the state-space equation and use a holistic Model-Predictive Control approach in which quadratic programming is involved for optimization. On top of this, they use a Multi-Model Adaptive Predictive controller (MMAPC) to account for the uncertainties in cornering stiffness of the tires and adhesive conditions. They simulate the results using MATLAB/Simulink and Carsim. They compare results of the holistic MPC with and without the MMAPC. They also compare the results of the holistic MPC with traditional heirarchical MPCs and a controller with seperate structure. We are using MATLAB for the simulations.

Index Terms

Autonomous vehicles, path following control, holistic control structure, model predictive control, adaptive control, multiple-model theory.

I. Introduction

An autonomous vehicle, or a driverless vehicle, is one that is able to operate itself and perform necessary functions without any human intervention, through ability to sense its surroundings. Autonomous vehicles can reduce the number of car crashes, provide a more convenient mode of transport to people who cannot drive, reduce driving fatigue and make supply chain and logistics much faster. Path following is one of the key aspects of autonomous driving. It ensures that the vehicle follows a particular path/lane which is very crucial to avoid accidents and obey lane rules. Many approaches like PID, Fuzzy Control, Sliding Mode Control, Optimal Control, MPC, Gain-Scheduling, H-∞, μ-synthesis have been discussed in the past by various researchers for the path-following problem. Out of these, MPC has been found to be the most promising for its ability to deal with multiple constraints, adapt to various scenarios and allow nonlinearity of the system.

Traditionally, heirarchical controllers are used for path-following, where, an upper-level controller calculates the steering angle and generalized forces, while, a lower-level allocator allocates the control commands to the wheels. This approach performs well in most driving conditions but fails in extreme conditions. This is because the forces calculated by the upper-level controller are not fulfilled by the actuators. For such extreme conditions, a holistic controller performs

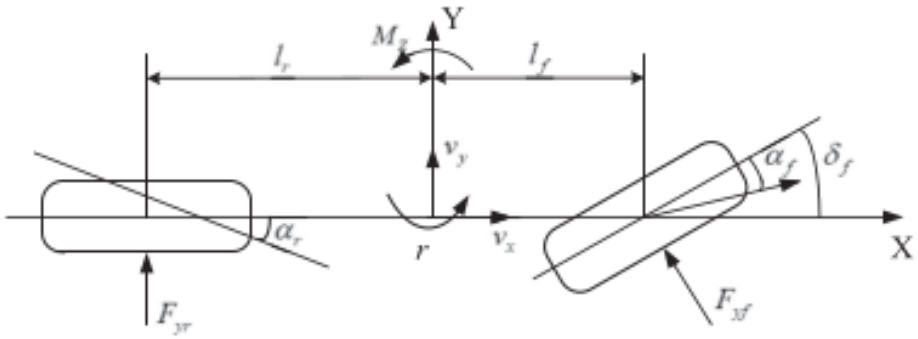

Fig. 1. Simplified Single Track Vehicle Model [1]

better due to direct calculation of the control commands without intermediate variables. Holistic controllers usually require complex dynamic models.

However, in the approach proposed in the journal paper [1], the lateral and longitudinal motions and the steering angle and torque commands on the steering wheels are decoupled to reduce the computational complexity and burden of the MPC. Also, a weight adaptive mechanism is used to coordinate the longitudinal motions and regulate the slip ratio of each wheel. For dealing with the uncertainties in cornering stiffness of tires, a blending-based adaptive law based on multi-model theory is used. This increases accuracy of the internal model.

II. System Modelling and State Space Equation

In the paper [1], a single track model (Fig. 1) of the vehicle is introduced based on the following assumptions: a) The steering angle of the wheels is small, b) The difference of steering angle between left and right wheels caused by Ackermann steering geometry is neglected. [4]

Using this model, we can write the Force balance along y direction as,

$$m\dot{v_y} = F_{yr} + F_{yf}cos(\delta_f) + F_{coriollis} \tag{1}$$

, where,

m is mass of the vehicle, v_y is lateral velocity of the vehicle, F_{yr} and F_{yf} are cornering forces on front and rear tires given by F_yr = $C_r\alpha_r$ and F_yf = $C_f\alpha_f$, where, α_r and α_f are rear and front wheel cornering angles. and C_f and C_r are front and rear wheel Cornering Stiffnesses. $F_{coriollis}$ is the coriollis force on the body given as $-mv_xr$, where, v_x is longitudinal velocity of

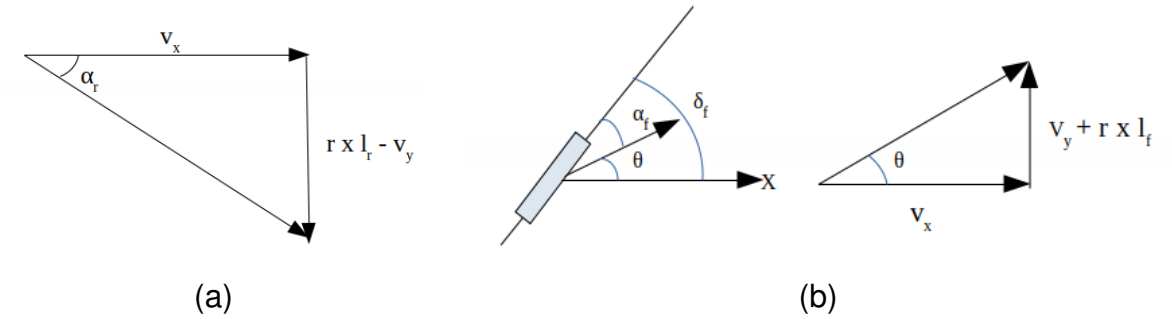

Fig. 2. Calculation of (a) α_r and (b) α_f

the vehicle and r is the yaw rate. δ_f is the steering angle of the vehicle in radians. As steering angle δ_f is small, cos(δ_f) ≈ 1.

If we draw the velocity vector diagrams as in Fig. 2, we realize that

(a) From Fig. 2a, $\alpha_r \approx tan(\alpha_r) = \dfrac{(rl_r - v_y)}{v_x}$ since α_r and α_f are small.

(b) From Fig. 2b, $\alpha_f = \delta_f - \theta$ and $\theta \approx tan(\theta) = \dfrac{(v_y + rl_f)}{v_x} \Rightarrow \alpha_f = \delta_f - \dfrac{(v_y + rl_f)}{v_x}$

Thus, substituting all these into Eq. (1),

$$\dot{v_y} = \frac{1}{m} * \left[C_r \left(\frac{rl_r - v_y}{v_x} \right) + C_f \left(\delta_f - \left(\frac{v_y + rl_f}{v_x} \right) \right) - m v_x r \right] \tag{2}$$

By balancing torques in the single track model (Fig. 1), we can write:

$$\begin{aligned} I_z \dot{r} &= F_{yf} l_f + M_z - F_{yr} l_r \\ \dot{r} &= \frac{1}{I_z} \left[l_f C_f \left(\delta_f - \left(\frac{v_y + rl_f}{v_x} \right) \right) + M_z - l_r C_r \left(\frac{rl_r - v_y}{v_x} \right) \right] \end{aligned} \tag{3}$$

,where, M_z is the active yaw moment.

Since δ_f is very small, we can assume F_{yf} to be perpendicular to the length l_f. We can write the equations Eq. (2) and Eq. (3) in state-space form as:

$$\begin{bmatrix} \dot{v_y} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\dfrac{C_f + C_r}{m v_x} & -v_x - \left(\dfrac{l_f C_f - l_r C_r}{m v_x} \right) \\ -\dfrac{C_f l_f - C_r l_r}{I_z v_x} & -\dfrac{C_f l_f^2 + C_r l_r^2}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \dfrac{C_f}{m} & 0 \\ \dfrac{l_f C_f}{I_z} & \dfrac{1}{I_z} \end{bmatrix} \begin{bmatrix} \delta_f \\ M_z \end{bmatrix} \tag{4}$$

From Fig. 3, applying torque balance, we can write:

$$\begin{aligned} M_z &= -F_{xlf} \frac{W}{2} + F_{xrf} \frac{W}{2} - F_{xlr} \frac{W}{2} + F_{xrr} \frac{W}{2} \\ &\approx -\frac{T_{xlf} cos(\delta_f)}{r_w} \frac{W}{2} + \frac{T_{xrf} cos(\delta_f)}{r_w} \frac{W}{2} - \frac{T_{xlr}}{r_w} \frac{W}{2} + \frac{T_{xrr}}{r_w} \frac{W}{2} \end{aligned} \tag{5}$$

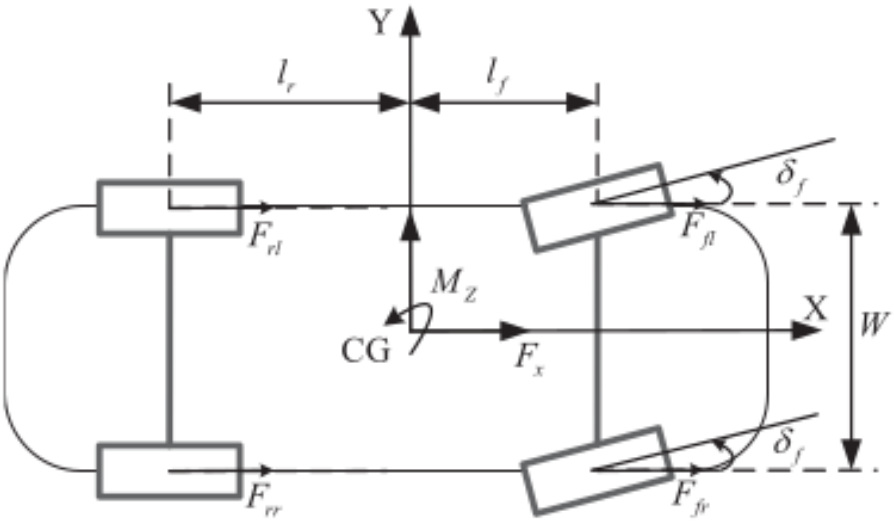

Fig. 3. Extended Single Track Vehicle Model [1]

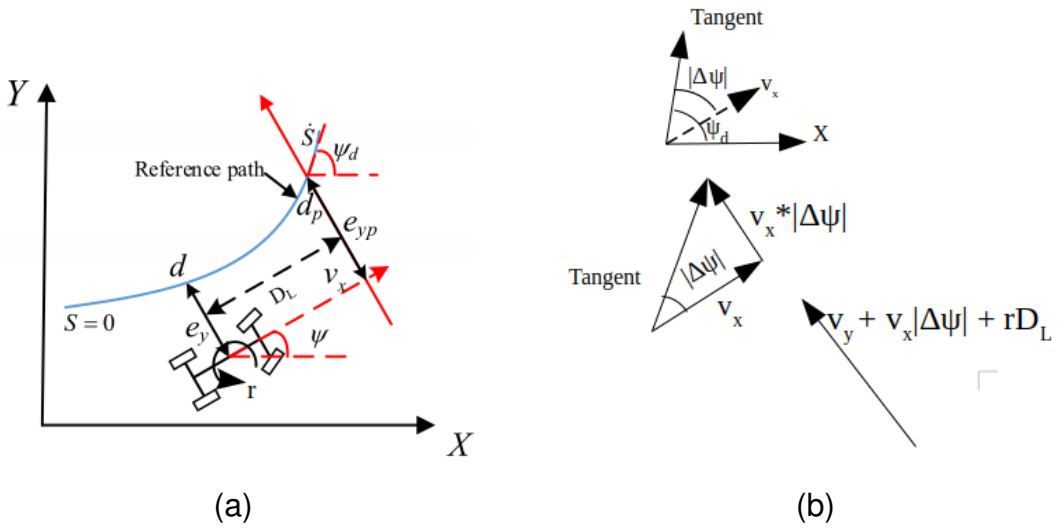

Fig. 4. (a) Path following model [1] and (b) Velocity vector diagrams to calculate $\dot{e_{yp}}$

, where, F_{xlf}, F_{xrf}, F_{xlr} and F_{xrr} are components of tire forces in longitudinal direction on left front, right front, left rear and right rear wheels respectively. As $sin(\delta_f)$ is small, lateral components of these forces are neglected. T_{xlf}, T_{xrf}, T_{xlr} and T_{xrr} are the torques on the wheels. r_w is the wheel radius.

In Figure 4, d represents the point closest to the vehicle on the curve and e_y represents the lateral error between the Center of Gravity of the car and the closest point d on the path. D_L

is the preview distance and e_{yp} is the lateral error from center of gravity to the preview point d_p on the desired path. ψ_d is the desired heading angle and ψ is the actual heading angle. The heading error is defined in [1] by $\Delta\psi = \psi - \psi_d$. From the vector diagrams in Fig. 4b, we realize that, $v_x tan(|\Delta\psi|) \approx v_x|\Delta\psi|$ gives us a velocity perpendicular to v_x, i.e., in lateral direction. We know that v_y is along lateral direction and the velocity rD_L due to yaw rate r is also along the lateral direction. $\dot{\psi}$ is basically the yaw rate r of the vehicle and $\dot{\psi}_d$ is equal to the tangential velocity along the curve which can be written as $\rho(d)v_x$, where, $\rho(d)$ is the curvature of the curve at point d, i.e., $\left(\frac{1}{radius\ of\ curvature\ at\ point\ d}\right)$. Thus, we can write the following equations:

$$\begin{aligned} \dot{e_{yp}} &= v_x\Delta\psi + v_y + rD_L \\ \dot{\Delta\psi} &= \dot{\psi} - \dot{\psi}_d = r - \rho(d)v_x \end{aligned} \tag{6}$$

The preview distance D_L is a function of v_x defined as:

$$D_L = \begin{cases} D_{low} & v_x \ \leq \ v_{xthel} \\ v_x\Delta t & v_{xthel} \ < \ v_x \ < \ v_{xtheh} \\ D_{high} & v_x \ \geq \ v_{xtheh} \end{cases} \tag{7}$$

, where, Δt is the preview time, which is set to 0.4 s. D_{low} and D_{high} are lower and upper thresholds of D_L respectively set to 2 m and 12 m. v_{xthel} and v_{xtheh} are the lower and upper thresholds of longitudinal velocity respectively set to 5 m/s and 30 m/s. Combining equations Eq. (4) and Eq. (6), we can write,

$$\begin{aligned} \dot{e_{yp}} &= v_x\Delta\psi + v_y + rD_L \\ \dot{\Delta\psi} &= r - \rho(d)v_x \\ \dot{v_y} &= -\left(\frac{C_f + C_r}{mv_x}\right)v_y + \left(-v_x - \left(\frac{l_fC_f - l_rC_r}{mv_x}\right)\right)r + \frac{C_f}{m}\delta_f \\ \dot{r} &= -\left(\frac{l_fC_f - l_rC_r}{I_zv_x}\right)v_y - \left(\frac{C_fl_f^2 + C_rl_r^2}{I_zv_x}\right)r + \frac{l_fC_f}{I_z}\delta_f + \frac{1}{I_z}M_z \end{aligned} \tag{8}$$

To avoid complex coupling relationships that will increase nonlinearity and computatinal burden, we consider a simplified model for the velocity tracking problem:

$$\begin{aligned} e_x &= v_x - v_{xd} \\ \dot{e_x} &= a_x - a_{xd} = \frac{F_{xt}}{m} - a_{xd} \end{aligned} \tag{9}$$

,where, v_{xd} is the desired longitudinal velocity, a_{xd} is desired longitudinal acceleration, m is mass of vehicle and F_{xt} is the total longitudinal force on the body which can be expressed as:

$$\begin{aligned} F_{xt} &= F_{xlf} \ + \ F_{xrf} \ + \ F_{xlr} \ + \ F_{xrr} \\ &\approx \frac{T_{xlf} cos(\delta_f)}{r_w} + \frac{T_{xrf} cos(\delta_f)}{r_w} + \frac{T_{xlr}}{r_w} + \frac{T_{xrr}}{r_w} \end{aligned} \tag{10}$$

Combining equations Eq.(5) with Eq.(6) - (10) and writing in state-space form, the state vector can be defined as $x = [e_x \ e_{yp} \ \Delta\psi \ v_y \ r]^T$, the control input is defined as $u = [\delta_f \ F_{xlf} \ F_{xrf} \ F_{xlr} \ F_{xrr}]^T$, the disturbance is defined as $w = [-a_{xd} \ v_x\rho(d) \ 0 \ 0 \ 0]^T$ and the output of the system is defined as $y = [e_x \ e_{yp} \ \Delta\psi \ v_y]^T$. The continuous time state-space form of the combined system can be written as follows:

$$\begin{aligned} \dot{x} &= Ax \ + \ Bu \ + \ w \\ y &= Cx \end{aligned} \tag{11}$$

, where, $A \in \Re^{\ 5 \ \times \ 5}$, $B \in \Re^{\ 5 \ \times \ 5}$ and $C \in \Re^{\ 4 \ \times \ 5}$ are given as:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_x & 1 & D_L \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\dfrac{C_f + C_r}{m v_x} & -v_x - \dfrac{C_f l_f - C_r l_r}{m v_x} \\ 0 & 0 & 0 & \dfrac{C_r l_r - C_f l_f}{I_z v_x} & -\dfrac{C_f l_f^2 + C_r l_r^2}{I_z v_x} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & \dfrac{1}{m} & \dfrac{1}{m} & \dfrac{1}{m} & \dfrac{1}{m} \\ 0 & 0 & v_x & 1 & D_L \\ 0 & 0 & 0 & 0 & 1 \\ \dfrac{C_f}{m} & 0 & 0 & 0 & 0 \\ \dfrac{C_f l_f}{I_z} & -\dfrac{W}{2I_z} & \dfrac{W}{2I_z} & -\dfrac{W}{2I_z} & \dfrac{W}{2I_z} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

By taking e_x as the state of the system, the longitudinal and lateral motions can be decoupled, thus reducing the computational burden [1].

III. Controller Design

This section discusses the design of 2 control algorithms: (a) Holistic MPC and (b) Multi-model Adaptive Predictive controller with adaptive scheme for cornering stiffness uncertainties.

A. Holistic Model Predictive Path Following Controller

First, we discretize Eq. (11) with respect to time. For this, we assume that the state variable changes from $x(k)$ to $x(k+1)$ in time step of $\Delta T = 0.01secs$. Thus $\dot{x}$ can be written as $\frac{x(k+1)-x(k)}{\Delta T}$. Thus, equation (11) becomes,

$$\begin{aligned} x(k+1) &= x(k) + A\Delta T x(k) + B\Delta T u(k) + \Delta T w(k) \\ x(k+1) &= A_d x(k) + B_d u(k) + \Delta T w(k) \\ y(k) &= C_d x(k) \end{aligned} \tag{12}$$

, where, $A_d = I + A\Delta T$, $B_d = B\Delta T$ and $C_d = C$.

Now, taking one more time step and substituting value of $x(k+1)$, we get:

$$\begin{aligned} x(k+2) &= A_d x(k+1) + B_d u(k+1) + \Delta T w(k+1) \\ &= A_d(A_d x(k) + B_d u(k) + \Delta T w(k)) + B_d u(k+1) + \Delta T w(k+1) \\ &= A_d^2 x(k) + A_d B_d u(k) + B_d u(k+1) + A_d \Delta T w(k) + \Delta T w(k+1) \end{aligned} \tag{13}$$

N_p is no. of steps in prediction horizon, i.e., time into the future till where we predict from current time and N_c is no. of steps in control horizon, i.e., time into the future until which changes in control inputs is allowed. Similar to Eq.(13), substituting values of previous x in

equation for x of current time step, we get the following equations:

$$
\begin{aligned}
x(k+3) &= A_d^3 x(k) + A_d^2 B_d u(k) + A_d B_d u(k+1) + B_d u(k+2) \\
&\quad + A_d^2 \Delta T w(k) + A_d \Delta T w(k+1) + \Delta T w(k+2) \\
x(k+4) &= A_d^4 x(k) + A_d^3 B_d u(k) + A_d^2 B_d u(k+1) + A_d B_d u(k+2) + B_d u(k+3) \\
&\quad + A_d^3 \Delta T w(k) + A_d^2 \Delta T w(k+1) + A_d \Delta T w(k+2) + \Delta T w(k+3) \\
&\quad . \\
&\quad . \\
x(k+N_c) &= A_d^{N_c} x(k) \\
&\quad + [A_d^{N_c-1} B_d u(k) + A_d^{N_c-2} B_d u(k+1) + .. + A_d B_d u(k+N_c-2) + B_d u(k+N_c-1)] \\
&\quad + [A_d^{N_c-1} \Delta T w(k) + A_d^{N_c-2} \Delta T w(k+1) + .. + \Delta T w(k+N_c-1)] \\
x(k+N_c+1) &= A_d x(k+N_c) + \Delta T w(k+N_c) \\
&\quad [Since\ N_c\ is\ defined\ as\ max.\ time\ steps\ until\ which\ control\ input\ is\ allowed] \\
&\quad = A_d^{N_c+1} x(k) \\
&\quad + [A_d^{N_c} B_d u(k) + A_d^{N_c-1} B_d u(k+1) + .. + A_d^2 B_d u(k+N_c-2)] + [A_d B_d u(k+N_c-1)] \\
&\quad + [A_d^{N_c} \Delta T w(k) + A_d^{N_c-1} \Delta T w(k+1) + .. + A_d \Delta T w(k+N_c-1) + \Delta T w(k+N_c)] \\
x(k+N_c+2) &= A_d^{N_c+2} x(k) \\
&\quad + [A_d^{N_c+1} B_d u(k) + A_d^{N_c} B_d u(k+1) + .. + A_d^3 B_d u(k+N_c-2)] + [A_d^2 B_d u(k+N_c-1)] \\
&\quad + [A_d^{N_c+1} \Delta T w(k) + A_d^{N_c} \Delta T w(k+1) + .. + A_d^2 \Delta T w(k+N_c-1) + .. + \Delta T w(k+N_c+1)] \\
&\quad \vdots \\
&\quad . \\
x(k+N_p) &= A_d^{N_p} x(k) \\
&\quad + [A_d^{N_p-1} B_d u(k) + A_d^{N_p-2} B_d u(k+1) + .. + A_d^{N_p-N_c+1} B_d u(k+N_c-2)] \\
&\quad + [A_d^{N_p-N_c} B_d u(k+N_c-1)] \\
&\quad + [A_d^{N_p-1} \Delta T w(k) + A_d^{N_p-2} \Delta T w(k+1) + .. + A_d^{N_p-N_c} \Delta T w(k+N_c-1) + .. \\
&\quad + \Delta T w(k+N_p-1)]
\end{aligned}
\tag{14}
$$

Note: (a) In [1], the last w term of $x(k+N_c)$ equation is $\Delta Tw(k+N_p)$ which is not possible because you do not know the value of disturbance at a future time step in current time step ($N_p > N_c$). Hence, it is a printing error in the paper. The correct equation should be as we have derived in Eq.(14).

(b) In [1], the last w term of the $x(k+N_p)$ equation is $\Delta Tw(k+N_c-1)$ but by our calculations, as you can substitute $x(k+i)$ values and see, we are getting it as $w(k+N_p-1)$. This is another printing error in the journal paper.

Thus, we can write Eq.(12) - (14) in matrix form as:

$$Y(k) = C_p x(k) + D_p U(k) + E_p W(k) \tag{15}$$

, where, $x(k) \in \Re^{\ 5 \ \times \ 1}$

$$Y(k) = [y(k+1); y(k+2); \cdots ; y(k+N_p)]^T \in \Re^{\ 4N_p \ \times \ 1}$$

$$U(k) = [u(k); u(k+1); \cdots ; u(k+N_c-1)]^T \in \Re^{\ 5N_c \ \times \ 1}$$

$$W(k) = [\Delta Tw(k); \Delta Tw(k+1); \cdots ; \Delta Tw(k+N_p-1)]^T \in \Re^{5N_p \ \times \ 1}$$

$$C_p = \begin{bmatrix} C_d A_d \\ C_d A_d^2 \\ . \\ . \\ C_d A_d^{N_p} \end{bmatrix} \in \Re^{4N_p \ \times \ 5}$$

$$D_p = \begin{bmatrix} C_d B_d & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ C_d A_d^{N_c-1} B_d & C_d A_d^{N_c-2} B_d & \cdots & C_d A_d B_d & C_d B_d \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ C_d A_d^{N_p-1} B_d & C_d A_d^{N_p-2} B_d & \cdots & C_d A_d^{N_p-N_c+1} B_d & C_d A_d^{N_p-N_c} B_d \end{bmatrix} \in \Re^{4N_p \ \times \ 5N_c}$$

$$E_p = \begin{bmatrix} C_d & 0 & \cdots & 0 & 0 \\ C_d A_d & C_d & \cdots & 0 & 0 \\ & & \vdots & & \\ C_d A_d^{N_p-1} & C_d A_d^{N_p-2} & \cdots & C_d A_d & C_d \end{bmatrix} \in \Re^{4N_p \ \times \ 5N_p}$$

The optimization function of the MPC is given as follows:

$$minJ = (Y - Y_{des})^T Q (Y - Y_{des}) + U^T R U \tag{16}$$

,where, $Q \in \Re^{4N_p \times 4N_p}$ and $R \in \Re^{5N_c \times 5N_c}$ are weight matrices corresponding to the error in output $(Y - Y_{des})$ and control input $U \in \Re^{5N_c \times 1}$ respectively. $Y_{des} \in \Re^{4N_p \times 1}$ is desired system output.

$$Q = \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & \ddots & \\ & & & Q_{N_p} \end{bmatrix} R = \begin{bmatrix} R_1 & & & \\ & R_2 & & \\ & & \ddots & \\ & & & R_{N_c} \end{bmatrix}$$

$$Q_i = \begin{bmatrix} W_{ex} & & & \\ & W_{eyp} & & \\ & & W_{\Delta\psi} & \\ & & & W_{v_y} \end{bmatrix} R_i = \begin{bmatrix} W_{\delta_f} & & & & \\ & W_{Ffl} & & & \\ & & W_{Ffr} & & \\ & & & W_{Frl} & \\ & & & & W_{Frr} \end{bmatrix}$$

$$Y_{des} = \begin{bmatrix} Y_{des1} \\ Y_{des2} \\ \vdots \\ Y_{desN_p} \end{bmatrix}, \; Y_{desi} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \; since\ we\ would\ like\ all\ errors\ and\ lateral\ velocity\ to\ be\ zero$$

, where, W_{ex}, W_{eyp}, $W_{\Delta\psi}$ and W_{v_y} are the weights corresponding to longitudinal error for velocity tracking, lateral error, heading error and lateral velocity respectively. W_{δ_f} is the weight for steering angle and W_Fij are the weights for the longitudinal forces on the wheels.

For improving vehicle performance under various driving conditions, especially some extreme situations, an adaptive mechanism is proposed for some of the weights.

For the weighing matrix on the system output, an adaptive mechanism is designed for the weight on velocity tracking (longitudinal error) according to hyperbolic tangent function as follows: (Fig. 5a)

$$Q_s = max\left(\frac{|e_y|}{e_{ythe}}, \frac{|\Delta\psi|}{\Delta\psi_{the}}, \frac{|\beta|}{\beta_{the}}\right)$$

$$W_{ex} = \begin{cases} W_{ex0} & if\ Q_s \leq 1 \\ a_s + b_s tanh\left(\frac{k_s}{Q_s}\right)^{c_s} & otherwise \end{cases} \tag{17}$$

, where, e_{ythe}, $\Delta\psi_{the}$, β_{the} are the defined thresholds for the lateral error, heading error and side slip angle β respectively. Side slip angle is calculated as $\beta \approx \frac{v_y}{v_x}$. W_{ex0} is the origin weight of

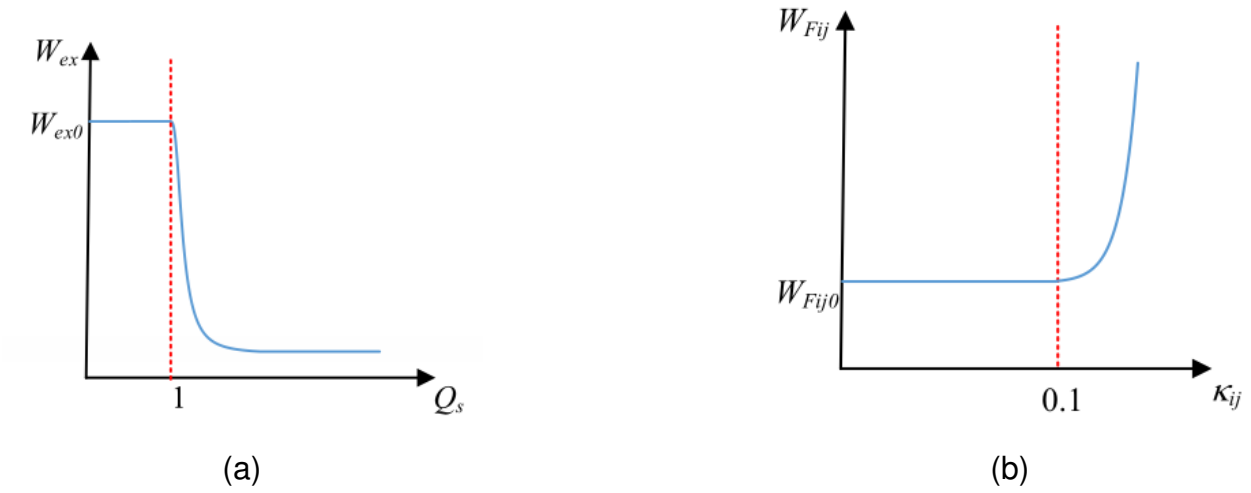

Fig. 5. Adaptive mechanisms for W_{ex} and W_{Fij} [1]

velocity tracking, a_s, b_s, k_s and c_s are parameters of the weight-adaptive mechanism, $tanh()$ is defined as:

$$tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$$

Here, the parameters a_s, b_s, k_s and c_s are tuned such that W_{ex} becomes small. This is because, when Qs becomes greater than 1, it means that controller has difficulty maintaining the path-tracking performance of dynamic stability. Thus, at such times, we would like to give lower priority to e_x. As seen in Fig. 5a, W_{ex} decreases from its origin value W_{ex0} over time. Since $tanh(s)$ is an increasing function, we need to select a negative value of b_s to reduce W_{ex}. a_s should be greater than $(-b_s)$ because maximum value of $tanh(s)$ is 1 and we wouldn't want to have a negative weight value.

An adaptive mechanism has been proposed in the journal paper [1] for the control longitudinal forces on the wheels to prevent oversized slip ratio. The mechanism is given as follows:

$$\begin{aligned} \kappa r_{ij} &= |\kappa_{ij}| \\ W_{Fij} &= \begin{cases} W_{Fij0} & if\ \kappa r_{ij} \ \leq \ 0.1 \\ W_{Fij0} * e^{k_w(\kappa r_{ij} - 0.1)} & otherwise \end{cases} \end{aligned} \tag{18}$$

, where, W_{Fij0} is the origin weight of longitudinal force on wheels, k_w is parameter of weight

adaptive mechanism, κ_{ij} is the slip ratio on each wheel which is calculated as:

$$\kappa_{ij} = \begin{cases} 1 - \dfrac{v_x}{r_w \omega_{ij}} & if\ v_x\ <\ r_w\omega_{ij} \\ \dfrac{r_w\omega_{ij}}{v_x} - 1 & if\ v_x\ >\ r_w\omega_{ij} \end{cases} \tag{19}$$

, where, r_w is the wheel radius and ω_{ij} is the angular wheel speed. As the absolute value of κ_{ij} becomes larger than 0.1, W_{Fij} will rise sharply from its origin value W_{Fij0}. Thus, the longitudinal forces on the wheels will be released rapidly thus preventing oversized slip ratio. This improves controllability of the autonomous vehicle.

The constraints on control output are defined as follows:

$$U_{min} \leq U \leq U_{max} \tag{20}$$

, where U_{min} and $U_{max} \in \Re^{5N_c\ \times\ 1}$ are the lower and upper bounds of the control input U respectively. They are defined as:

$$U_{min} = \begin{bmatrix} U_{min1} \\ U_{min2} \\ \vdots \\ U_{minN_c} \end{bmatrix} \quad U_{max} = \begin{bmatrix} U_{max1} \\ U_{max2} \\ \vdots \\ U_{maxN_c} \end{bmatrix}$$

$$U_{min1} = max(u_{min}, u_p - \Delta u_{max}) \quad U_{max1} = min(u_{max}, u_p + \Delta u_{max})$$

$$U_{min2..N_c} = u_{min} \quad U_{max2..N_c} = u_{max}$$

, where, u_{min} is the lower bound of control input at any time step, u_{max} is the upper bound for control input at any time step, u_p is the control input from the last calculation interval, Δu_{max} is the maximum allowable change in control input within the sampling time ΔT. Thus, the control input is not only constrained by the lower and upper bounds, but it is also constrained by the allowable variation from last control input so as to constrain the rate of change.

To ensure driving safety, constraints for system output are defined as follows with a slack variable ε to prevent the unsolvable problem:

$$Y_{min} - \varepsilon I_{4N_p*1}\ \leq\ Y\ \leq\ Y_{max} + \varepsilon I_{4N_p*1} \tag{21}$$

The cost function in Eq. (16) can be rewritten as:

$$\begin{aligned} &min\ J = (Y - Y_{des})^T Q (Y - Y_{des}) + U^T R U + \rho\varepsilon^2 \\ &s.t.\ U_{min}\ \leq\ U\ \leq\ U_{max} \\ &Y_{min} - \varepsilon I_{4N_p*1}\ \leq\ Y\ \leq\ Y_{max} + \varepsilon I_{4N_p*1} \end{aligned} \tag{22}$$

, where ρ is the weight for penalizing the slack variable.

Ignoring disturbances, $Y = C_p x(k) + D_p U(k)$. Thus, substituting this in the equation (22), we get:

$$
\begin{aligned}
&minJ = (C_p x(k) + D_p U - Y_{des})^T Q (C_p x(k) + D_p U - Y_{des}) + U^T R U + \rho \varepsilon^2 \\
&= (U^T D_p^T + (C_p x(k) - Y_{des})^T) Q (C_p x(k) + D_p U - Y_{des}) + U^T R U + \rho \varepsilon^2 \\
&[Since\ (A+B)^T = A^T + B^T\ and\ (AB)^T = B^T A^T] \\
&= (C_p x(k) - Y_{des})^T Q (C_p x(k) - Y_{des}) + U^T D_p^T Q (C_p x(k) - Y_{des}) + (C_p x(k) - Y_{des})^T Q D_p U + \\
&U^T D_p^T Q D_p U + U^T R U + \varepsilon \rho \varepsilon \\
&= (C_p x(k) - Y_{des})^T Q (C_p x(k) - Y_{des}) + 2 (C_p x(k) - Y_{des})^T Q D_p U + U^T (D_p^T Q D_p + R) U + \varepsilon \rho \varepsilon
\end{aligned}
\tag{23}
$$

Since $(C_p x(k) - Y_{des})^T \in \Re^{1 \times 4N_p}$, $Q \in \Re^{4N_p \times 4N_p}$,$D_p \in \Re^{4N_p \times 5N_c}$ and $U \in \Re^{5N_c \times 1}$, $(C_p x(k) - Y_{des})^T Q D_p U \in \Re^{1 \times 1}$. Thus the transpose of $(C_p x(k) - Y_{des})^T Q D_p U$ is same as itself.

Since $x(k)$ is the state known at the start of prediction horizon, it is constant during the optimization interval. C_p and Y_{des} are constant w.r.t. time. Thus, minimizing the cost in Eq. (23) is same as minimizing the cost without the $(C_p x(k) - Y_{des})^T Q (C_p x(k) - Y_{des})$ term. Eq. (23) can be written in Matrix form as:

$$
\begin{aligned}
minJ &= 2 \begin{bmatrix} (C_p x(k) - Y_{des})^T Q D_p & 0 \end{bmatrix} \begin{bmatrix} U \\ \varepsilon \end{bmatrix} + \frac{1}{2} * 2 \begin{bmatrix} U^T & \varepsilon \end{bmatrix} \begin{bmatrix} D_p^T Q D_p + R & 0 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} U \\ \varepsilon \end{bmatrix} \\
&= 2 \begin{bmatrix} (C_p x(k) - Y_{des})^T Q D_p & 0 \end{bmatrix} \begin{bmatrix} U \\ \varepsilon \end{bmatrix} + \frac{1}{2} * 2 \begin{bmatrix} U \\ \varepsilon \end{bmatrix}^T \begin{bmatrix} D_p^T Q D_p + R & 0 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} U \\ \varepsilon \end{bmatrix}
\end{aligned}
\tag{24}
$$

The constraints on U and Y can be combined with constraints on slack variable $0 \leq \varepsilon \leq \varepsilon_{max}$

and rewritten in matrix form as:

$$\begin{aligned}
&\begin{bmatrix} U_{min} \\ 0 \end{bmatrix} \leq \begin{bmatrix} U \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} U_{max} \\ \varepsilon_{max} \end{bmatrix} \; and \\
&Y_{min} - \varepsilon I_{4N_p*1} \leq C_p x(k) + D_p U \leq Y_{max} + \varepsilon I_{N_p*1} \\
&\Rightarrow D_p U - \varepsilon I_{4N_p*1} \leq Y_{max} - C_p x(k) \; and \\
&- D_p U - \varepsilon I_{N_p*1} \leq -Y_{min} + C_p x(k) \\
&Thus, \\
&\begin{bmatrix} D_p & -I_{4N_p*1} \\ -D_p & -I_{4N_p*1} \end{bmatrix} \begin{bmatrix} U \\ \varepsilon \end{bmatrix} \leq \begin{bmatrix} Y_{max} - C_p x(k) \\ -Y_{min} + C_p x(k) \end{bmatrix}
\end{aligned} \tag{25}$$

From Eq. (24) and Eq. (25), we can rewrite the cost function as in Quadratic Programming form, i.e., $min\ J\ =\ \frac{1}{2} x^T * H * x\ +\ f^T x$, as follows:

$$\begin{aligned}
min\ J &= \frac{1}{2} \hat{U}^T G_k \hat{U} + H_k \hat{U} \\
&or \\
min\ J &= \frac{1}{2} \hat{U}^T G_k \hat{U} + (H_k^T)^T \hat{U} \\
s.t.\ \hat{U}_{min} &\leq \hat{U} \leq \hat{U}_{max} \\
A_{cons} \hat{U} &\leq B_{cons}
\end{aligned} \tag{26}$$

, where,

$$G_k = 2 \begin{bmatrix} D_p^T Q D_p + R & 0 \\ 0 & \rho \end{bmatrix} \in \Re^{\ 5N_c+1\ \times\ 5N_c+1}$$

$$H_k = 2 \begin{bmatrix} (C_p x(k) - Y_{des})^T Q D_p & 0 \end{bmatrix} \in \Re^{\ 1\ \times\ 5N_c+1}$$

$$\hat{U} = \begin{bmatrix} U \\ \varepsilon \end{bmatrix} \; \hat{U}_{min} = \begin{bmatrix} U_{min} \\ 0 \end{bmatrix} \; \hat{U}_{max} = \begin{bmatrix} U_{max} \\ \varepsilon_{max} \end{bmatrix} \in \Re^{\ 5N_c+1\ \times\ 1}$$

$$A_{cons} = \begin{bmatrix} D_p & -I_{4N_p*1} \\ -D_p & -I_{4N_p*1} \end{bmatrix} \in \Re^{\ 8N_p\ \times\ 5N_c+1} \; B_{cons} = \begin{bmatrix} Y_{max} - C_p x(k) \\ -Y_{min} + C_p x(k) \end{bmatrix} \in \Re^{\ 8N_p\ \times\ 1}$$

Note: The dimensions have been mentioned a bit incorrectly in the paper and H_k is given incorrectly in the paper. In the constraints part in the paper [1], the second element of B_{cons} is given as $Y_{min}\ +\ C_p x(k)$ while it actually must be $-Y_{min}\ +\ C_p x(k)$.

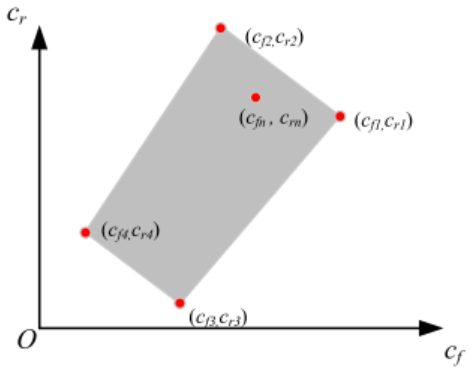

Fig. 6. Convex Polytope for Cornering Stiffness [1]

Note that the controller utilizes the longitudinal force on each wheel as the control output to each motor which is an indirect command. A simple transition must be made from force to torque before the command reaches the motor:

$$\begin{aligned} T_{xlf} &= \frac{F_{xlf} r_w}{cos(\delta_f)},\ T_{xrf} = \frac{F_{xrf} r_w}{cos(\delta_f)}, \\ T_{xlr} &= F_{xlr} r_w,\ T_{xrr} = F_{xrr} r_w \end{aligned} \tag{27}$$

These relations between force and torque are not accurate due to longitudinal-slip characteristics of tires, especially, when slip ratio is high. However, excessive slip ratio can be avoided because of the weight-adaptive mechanism proposed in Eq. (18). Thus, Eq. (27) has been considered to be acceptable in the research paper.

B. Multi-model Adaptive Predictive Controller

The vehicle system has a lot of uncertainties most of which come from cornering stiffnesses of tires. The cornering stiffness varies according to road adhesive conditions, vertical load and many other factors. To deal with these uncertainties, a polytope with 4 vertices $(C_{f1},\ C_{r1})$, $(C_{f2},\ C_{r2})$, $(C_{f3},\ C_{r3})$ and $(C_{f4},\ C_{r4})$ (Fig. 6) is considered to deal with all perturbations of cornering stiffnesses. $(C_{fn},\ C_{rn})$ is the calibration cornering stiffness. Based on the multiple-model theory [2], any uncertainties in the shadowed area in the polytope can be linearly expressed by the 4 vertices as:

$$\dot{x_p} = A_p(\eta)x_p(t) + B_p(\eta)u(t) = \sum_{i=1}^{4} w_i(t)[A_i x_p(t) + B_i u(t)] \tag{28}$$

, where, η is a parameter to denote the uncertainties in the model. $A_p \; and \; B_p$ denote the system matrices in Eq. (4), $w_i(t)$ is the weight of the corresponding model at time instant t which satisfies:

$$\sum_{i=1}^{4} w_i(t) = 1, \;\; w_i(t) \;\; \geq \;\; 0 \tag{29}$$

According to Eq. [**?**], we can see that estimation of $w_i(t)$ is equivalent to identification of the unknown matrices $A_p(\eta)$ and $B_p(\eta)$.By writing s as the differentiation operator $\frac{d}{dt}$ and multiplying $\frac{1}{s+\lambda}$ on both sides where λ is a constant design parameter, we can rewrite the state-space form as a linear parametric form. We do this conversion because this form is more suitable for the gradient descent method. For this identification problem, the classic form of the dynamics model can be writtem in terms of a linear parametric model, as follows [2] [3] :

$$\begin{aligned} z(t) &= \Theta_p(\eta)\Phi(t) \\ \Theta_p(\eta) &= [A_p(\eta) \;\; B_p(\eta)] \\ \Phi &= \frac{1}{s+\lambda}\begin{bmatrix} x_p \\ u \end{bmatrix}, \;\; z = \frac{s}{s+\lambda}x_p, \;\; \lambda \;\; > \;\; 0 \end{aligned} \tag{30}$$

Similarly, the fixed model of each vertex can be parametrized as:

$$\begin{aligned} z_i(t) &= \Theta_i\Phi(t) \\ \Theta_i &= [A_i \;\; B_i] \;\; i = 1, 2, ..., N \\ z_i &= \frac{s}{s+\lambda}x_i \end{aligned} \tag{31}$$

Then the estimation errors of the vertex model are defined as:

$$\varepsilon_i(t) = z(t) - \Theta_i\Phi(t), \quad i = 1, ..., N \tag{32}$$

Based on Eq.(28) and Eq. (29), the following relationships can be derived:

$$\begin{aligned} \Theta_p(t) &= \sum_{i=1}^{4} w_i(t)\Theta_i \\ \sum_{i=1}^{4} w_i(t)\varepsilon_i(t) &= 0 \end{aligned} \tag{33}$$

This can be expressed in matrix form as follows:

$$\begin{aligned} E(t)W(t) &= 0 \\ E(t) &= [\varepsilon_1(t)\varepsilon_2(t)\varepsilon_3(t)\varepsilon_4(t)] \in \Re^{\;1 \;\times\; 4} \\ W(t) &= [w_1(t)w_2(t)w_3(t)w_4(t)]^T \in \Re^{\;4 \;\times\; 1} \end{aligned} \tag{34}$$

From Eq.(28) and Eq.(34), the weight of the last model in the convex polytope can be calculated as $w_N = 1 - \sum_{i=1}^{N-1} w_i$. Thus, subtracting the error $\varepsilon_N(t)$ of the last model from both sides of Eq.(33), we get:

$$\begin{aligned}
\sum_{i=1}^{4} w_i(t)\varepsilon_i(t) - \varepsilon_4(t) &= -\varepsilon_4(t) \\
\sum_{i=1}^{4} w_i(t)\varepsilon_i(t) - \varepsilon_4(t)\sum_{i=1}^{4} w_i(t) &= -\varepsilon_4(t)\ [Since\ \sum_{i=1}^{4} w_i(t) = 1] \\
\sum_{i=1}^{3} w_i(t)(\varepsilon_i(t) - \varepsilon_4(t)) &= -\varepsilon_4(t) \\
E_R(t)W_R(t) &= -\varepsilon_4(t) \\
E_R(t) &= [(\varepsilon_1(t) - \varepsilon_4(t))...(\varepsilon_3(t) - \varepsilon_4(t))] \\
W_R(t) &= [w_1(t)w_2(t)w_3(t)]^T
\end{aligned} \tag{35}$$

Rewriting Eq. (35) and multiplying by E_R^T, we get:

$$E_R^T E_R W_R + E_R^T \varepsilon_4(t) = 0 \tag{36}$$

Now we are in a position to propose an adaptive identification law for weights of vertices.

Theorem. *Assume that the actual weights of the vertex models at time* t *are* $W_R(t)$*, the estimated weights are* $\hat{W}_R(t)$*, then the estimation error of weights can be expressed as* $\tilde{W}_R(t) = \hat{W}_R(t) - W_R(t)$*. This value can converge to zero if the estimated weights have the following Gradient-Descent based identification law:*

$$\begin{aligned}
\dot{\hat{W}}_R(t) &= -\Gamma E_R^T(t)E_R(t)W_R(t) - \Gamma E_R^T(t)\varepsilon_4(t) \\
\hat{w}_4(t) &= 1 - \sum_{i=1}^{3} \hat{w}_i(t)
\end{aligned} \tag{37}$$

, where, Γ *is a symmetric positive definite matrix to adjust the convergence speed.*

Proof. Substitute $\tilde{W}_R(t) = \hat{W}_R(t) - W_R(t)$ into Eq. (35), we get,

$$\begin{aligned}
E_R(t)[\hat{W}_R(t) - \tilde{W}_R(t)] &= -\varepsilon_4(t) \\
E_R(t)\tilde{W}_R(t) &= E_R(t)\hat{W}_R(t) + \varepsilon_4(t)
\end{aligned} \tag{38}$$

To prove the convergence ability of the proposed law, we define a Lyapunov function as:

$$V\left(\tilde{W}_R(t)\right) = \frac{1}{2}\tilde{W_R}^T(t)\Gamma^{-1}\tilde{W}_R(t) \tag{39}$$

Combining Eq. (37) and Eq. (38), we can write:

$$\begin{aligned}\dot{\tilde{W}}_R(t) = \dot{\hat{W}}_R(t) &= -\Gamma E_R^T(t)(E_R(t)\tilde{W}_R(t) - \varepsilon_4(t)) - \Gamma E_R^T(t)\varepsilon_4(t) \\ &= -\Gamma E_R^T(t)E_R(t)\tilde{W}_R(t)\end{aligned} \tag{40}$$

Differentiating the Lyapunov equation, we get:

$$\frac{dV\left(\tilde{W}_R(t)\right)}{dt} = \frac{1}{2}\ \dot{\tilde{W}}_R^T(t)\Gamma^{-1}\tilde{W}_R(t) + \frac{1}{2}\tilde{W}_R{}^T(t)\Gamma^{-1}\dot{\tilde{W}}_R(t)$$

$Substituting\ result\ from\ Eq.$(40) $here,\ we\ get:$

$$\frac{dV\left(\tilde{W}_R(t)\right)}{dt} = -\frac{1}{2}\tilde{W}_R^T(t)E_R^T(t)E_R(t)\Gamma^T\Gamma\tilde{W}_R(t) - \frac{1}{2}\tilde{W}_R{}^T(t)\Gamma^{-1}\Gamma E_R^T(t)E_R(t)\tilde{W}_R(t)$$

$$\frac{dV\left(\tilde{W}_R(t)\right)}{dt} = -\frac{1}{2}\tilde{W}_R^T(t)E_R^T(t)E_R(t)\tilde{W}_R(t) - \frac{1}{2}\tilde{W}_R{}^T(t)E_R^T(t)E_R(t)\tilde{W}_R(t)$$

$[Since\ \Gamma\ is\ symmetric\ positive\ definite,\ \Gamma^T = \Gamma^{-1}\ and\ hence\ \Gamma^T\Gamma = I,\ identity\ matrix]$

$$\frac{dV\left(\tilde{W}_R(t)\right)}{dt} = -\tilde{W}_R^T(t)E_R^T(t)E_R(t)\tilde{W}_R(t)$$

$$Thus,\ \frac{dV\left(\tilde{W}_R(t)\right)}{dt} = -\|E_R(t)\tilde{W}_R(t)\|^2\ \leq\ 0 \tag{41}$$

Thus, by Lyapunov global asymptotic stability theorem, since: a) $V\left(\tilde{W}_R(t)\right)$ is positive definite for $\tilde{W}_R(t) \neq 0$ and $V(0) = 0$ and b) $\frac{dV\left(\tilde{W}_R(t)\right)}{dt}\ <\ 0$ for $\tilde{W}_R(t) \neq 0$ and $\frac{dV(0)}{dt} = 0$, system is globally asymptotically stable and $\tilde{W}_R(t)$ converges to 0 as $t \longrightarrow \infty$

QED

However, although the proposed adaptive law in Eq. (37) has convergence ability and satisfies the 1st condition $\sum i = 1^4 w_i(t) = 1$ in Eq. (29), the second condition $w_i(t)\ \geq\ 0$, $i = 1, 2, ..., N$ is not guaranteed. Thus, a parameter projection model is introduced to the adaptive law. Firstly, the constraints in Eq. (29) are rewritten as:

$$S = \left\{\hat{W_R}\ \in\ \Re^3 | g(\hat{W_R})\ \leq\ 0\right\} \tag{42}$$

The function g() has the following definition to ensure convergence ability:

$$g(\eta) = -min\left\{\eta_1, \eta_2, \eta_3, 1 - \sum_{i=1}^{3}\eta_i\right\} \tag{43}$$

TABLE I

VEHICLE PARAMETERS USED IN SIMULATION

Symbol	Definition	Values
m	Vehicle mass	1650 kg
l_f	Distance from front axle to the center of mass	1.400 m
l_r	Distance from rear axle to the center of mass	1.650 m
I_z	Vehicle Moment of Inertia	3234 kgm^2
C_{f0}	Nominal Cornering Stiffness of Front tires	117000 N/rad
C_{r0}	Nominal Cornering Stiffness of Rear tires	117000 N/rad
Δt	Preview Time	0.4 s
ΔT	Time Step	0.01 s

, where, $\eta = [\eta_1, ..., \eta_{N-1}]$. Define S^0 and ∂S as the interior and boundary of S respectively. Then the adaptive law can be revised as follows by applying the gradient projection method [5]:

$$\begin{aligned} &If \ \left(\hat{W_R} \ \in S^0\right) \ or \ \left(\hat{W_R} \ \in \partial S \ and \ \hat{W_R}^T \nabla g \ \leq \ 0\right) \ : \\ &\dot{\hat{W}}_R = -\Gamma \left(E_R^T \varepsilon_4 + E_R^T E_R \hat{W}_R\right) \\ &Else: \\ &\dot{\hat{W}}_R = -\Gamma \left(E_R^T \varepsilon_4 + E_R^T E_R \hat{W}_R\right) + \Gamma \nabla g \left(\nabla g^T \Gamma \nabla g\right)^{-1} \nabla g^T \Gamma \left(E_R^T \varepsilon_4 + E_R^T E_R \hat{W}_R\right) \end{aligned} \tag{44}$$

, where, ∇g is the partial differential of g to the terms in η. For example, if η_1 is the minimum of η, then $g(\eta) = -\eta_1$ and ∇g should be $[-1, 0, ..., 0]_{N-1}$.

It is worth mentioning that compared with the robust methods, which are the classical way to deal with the uncertainties of tire cornering stiffness, the proposed multiple-model adaptive algorithm in the paper does not suffer from conservativeness.

IV. SIMULATION AND RESULTS

In the research paper, the performance of the proposed model is evaluated by simulations based on the Carsim-Simulink co-simulation platform. The control algorithm was coded in Matlab while Carsim provided the vehicle model of the E-class hatchback for the simulation. The parameters of the vehicle model are listed in Table I. The research paper provides two cases in the simulation which are as follows: (1) Show improvement brought by the Multiple Model Adaptive theory. (2) Verify the effectiveness and superiority of the proposed holistic controller

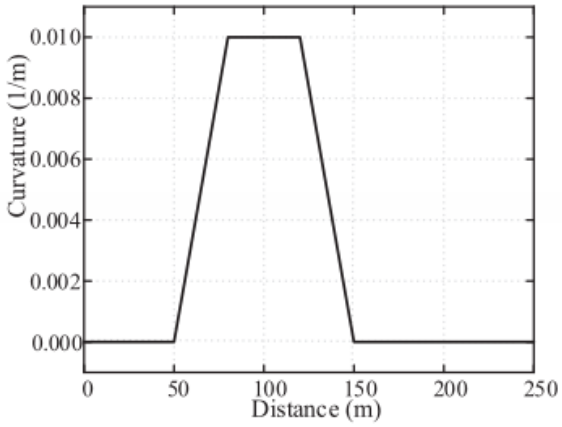

Fig. 7. Curvature of the J maneuver [1]

TABLE II
ROOT MEAN SQUARE VALUE COMPARISON

Controller	Lateral Error	Heading Error
MMAPC	0.03941	1.1542
MPC	0.04545	1.2162
Relative Ratio	86.7 %	94.9 %

A. Case One

In this case, the vehicle is made to follow a path with J-turn at a speed of 90 km/h with high adhesive condition. The road coefficient μ is 0.7 and the curvature of the path is shown in the Fig. 7. To show the improvement brought by the proposed model in the research paper, they compare it with a holistic model predictive controller without the multiple - model adaptive law.

The path following performance is measured using two metrics which are lateral error(m) and the heading error ($^\circ$). As shown in the Fig. 8 it is observed that both the controllers show satisfactory performance and the errors are constrained in a very small range which allows the vehicle to follow the reference path with a negligible difference. However, the proposed model shows a better performance than the nominal controller as the multiple model adaptive law enhances the internal accuracy of the controller.

To demonstrate the results in a better way, the quantitative analysis is introduced which utilizes the Root Mean Square (RMS) value. The RMS values of lateral errors and head errors are compared in Table II which clearly shows the improvement brought by the Multiple Model

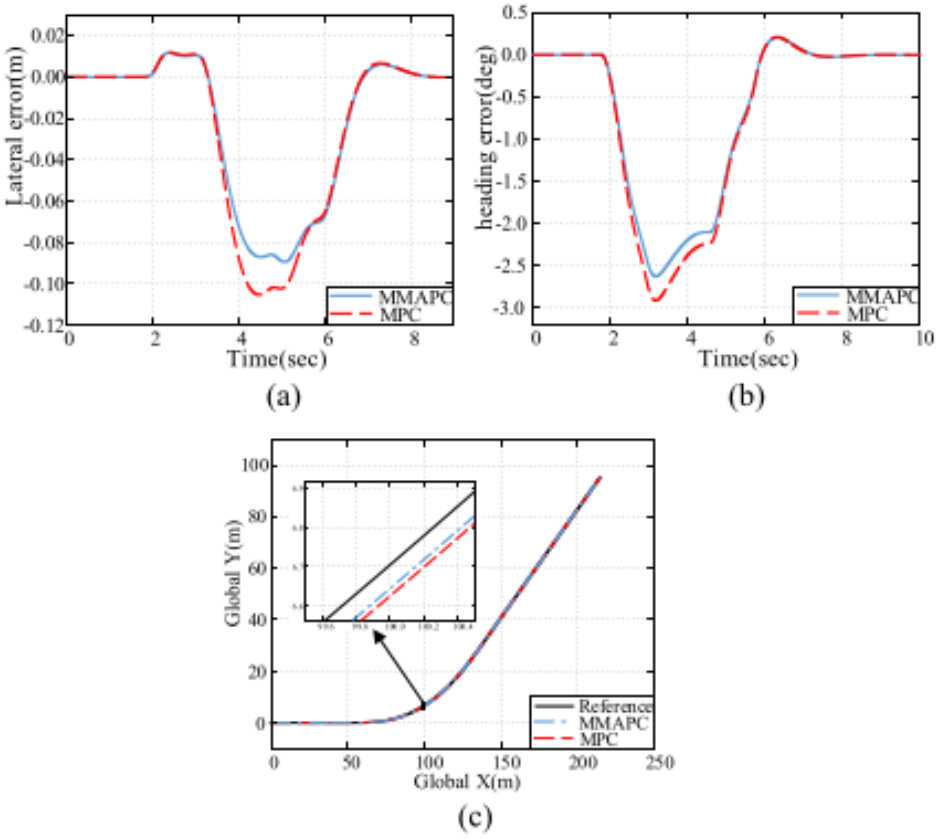

Fig. 8. Path following Performance (a) Lateral Error, (b) Heading Error, (c) Global Trajectory [1]

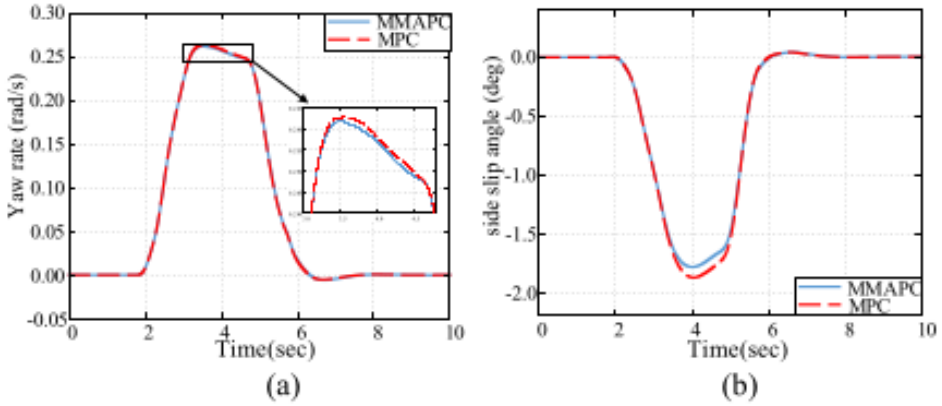

Fig. 9. Dynamic responses of (a) Yaw Rate, (b) Side Slip Angle [1]

Adaptive method. The Relative Ratio is the ratio of the error obtained by MMAPC to that obtained by MPC.

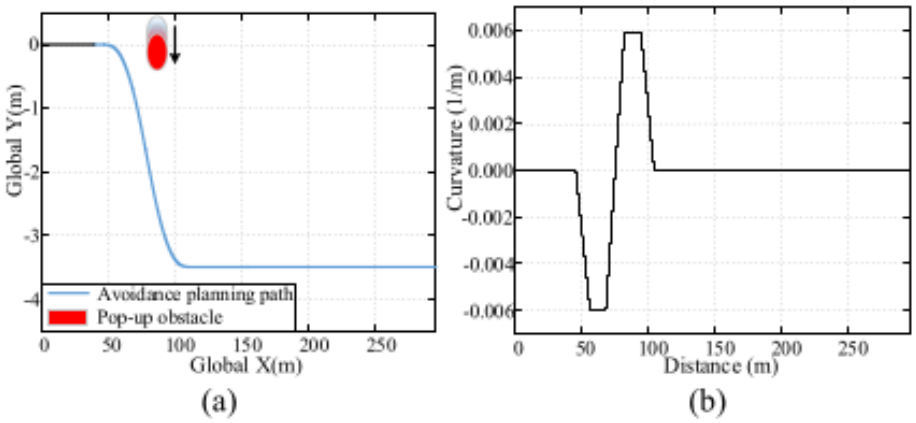

Fig. 10. (a) Reference Path and (b) Curvature [1]

On comparing the dynamic parameters like yaw rate and side slip angle as shown in Fig. 9 it is observed that the values are in reasonable ranges but the side slip angle of the adaptive predictive controller is a bit smaller than the nominal controller. Since the maneuver conducted in the scenario is very aggressive as compared to the real-life driving condition, it can be concluded that the performance of the proposed model on normal conditions has been verified.

B. Case Two

Now, to evaluate the effectiveness and features of the proposed model, an extreme driving condition is used in this case with the following details: (1) The vehicle is travelling at a very high speed of 120 km/h (2) The road is slippery with a road coefficient μ of 0.3 (3) An avoidance lane-change path is planned for collision avoidance when an obstacle (red/gray ellipse) suddenly appears from the left side as shown in the Fig. 10 (a). The curvature of the reference path is shown in the Fig. 10 (b).

The proposed model is compared with two other controllers. The first one is an integrated controller with a classical hierarchical structure in which the upper-level controller calculates the steering angle and the two generalized forces (F_{xt} and M_z) while a lower-level tire force allocator allocates the generalized forces to each tire. The second one is a controller with a separate structure in which the steering angle and the generalized forces are calculated separately, and there is also a lower-level allocator for force distribution.

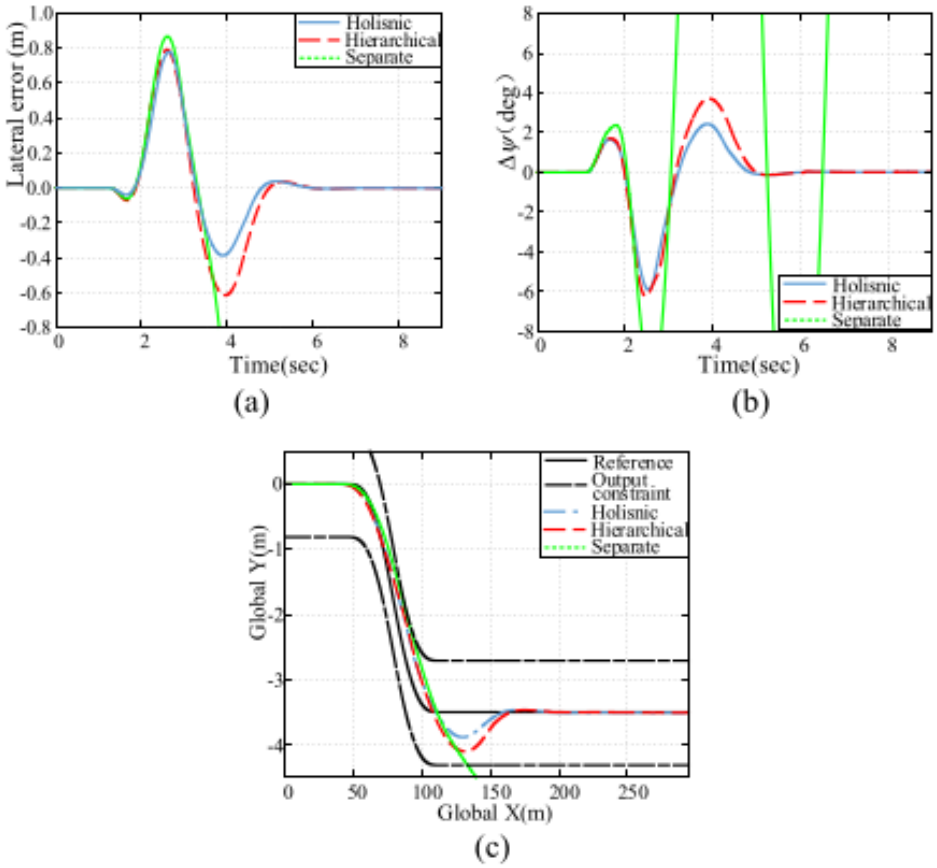

Fig. 11. Path Following Performance: (a) Lateral Error, (b) Heading Error, (c) Global Trajectory [1]

The path following performance of all these three controllers is shown in Fig. 11. Since the driving condition is very extreme it is evident from the figure that it is impossible to track the reference path with high accuracy. The controller with separate structure fails to track the reference path and loses its controllability. The controllers with holistic and hierarchical structure are still able to complete the path following objective and the errors of both the controllers are within the constraints. However it is observed that the holistic controller performs better than the hierarchical controller since the values of lateral and heading error are smaller. Considering the longitudinal slip characteristics in extreme conditions, it is difficult for the actuators to fulfill the forces calculated by the upper-level controller in the hierarchical model. This problem is solved by the holistic controller which calculates the control output on each corner without the generalized forces thus improving the performance. The RMS value comparison of this scenario

TABLE III

ROOT MEAN SQUARE VALUE COMPARISON

Controller	Lateral Error	Heading Error
Holistic	0.281	1.765
Heirarchical	0.328	1.963
Relative Ratio	85.6 %	89.7 %

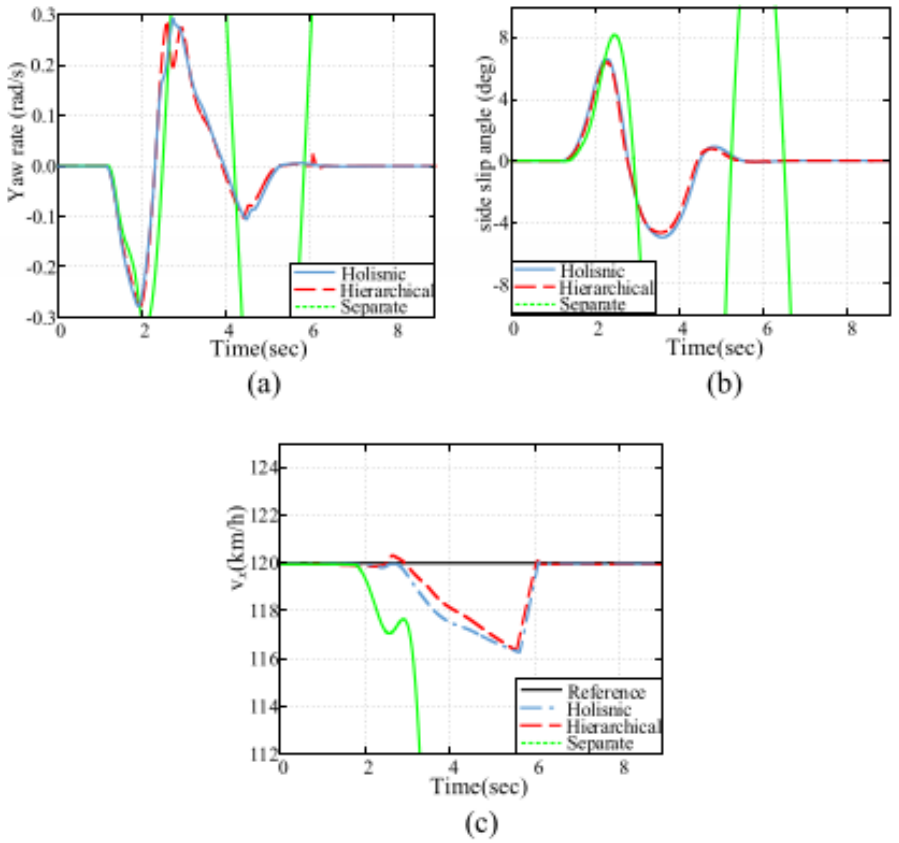

Fig. 12. Dynamic Responses: (a) Yaw Rate, (b) Lateral Velocity, (c) Longitudinal Velocity [1]

shown in Table III also shows that the proposed model performs better than the hierarchical model. Since the controller with separate structure fails to complete this task, it is not considered for this comparison.

The dynamic responses of the vehicle like the yaw rate, side slip angle and longitudinal velocity are shown in Fig. 12. It can be clearly observed from the figure that the controller with

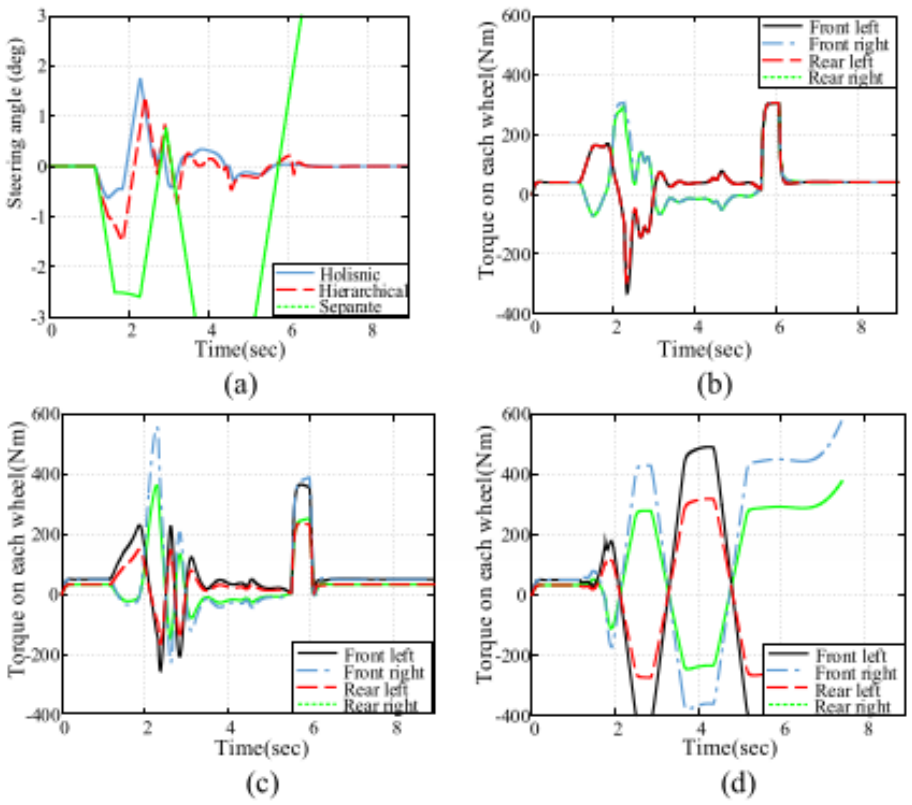

Fig. 13. Control Output: (a) Steering Angle, (b) Torques of Holistic Controller, (c) Torques of Heirarchical Controller, (d) Torques of Seperate Controller [1]

separate structure has lost its dynamic stability whereas the controllers with hierarchical and holistic structure though have relatively large slip angles as compared to case one, the values are still reasonable assuring the dynamic stability of the vehicle. Additionally, because of the weight adaptive mechanism, the longitudinal velocity deviates from the reference value when the lateral controller is not able to fulfill the task and once the extreme maneuver has been finished for a time lag, the velocity can be quickly restored to its reference value. The control output results are shown in Fig. 13. Since the separate structure based controller fails to complete this extreme task, the controls outputs are oversized and the vehicle goes out of control. The control outputs of hierarchical and holistic controllers are in reasonable range and it can be concluded that the holistic controller performs better as the torque outputs are smaller and much smoother.

The results of the slip ratio at each corner are shown in Fig. 14. Once again, the slip ratios of separate structure based controller are out of control. The controller based on hierarchical

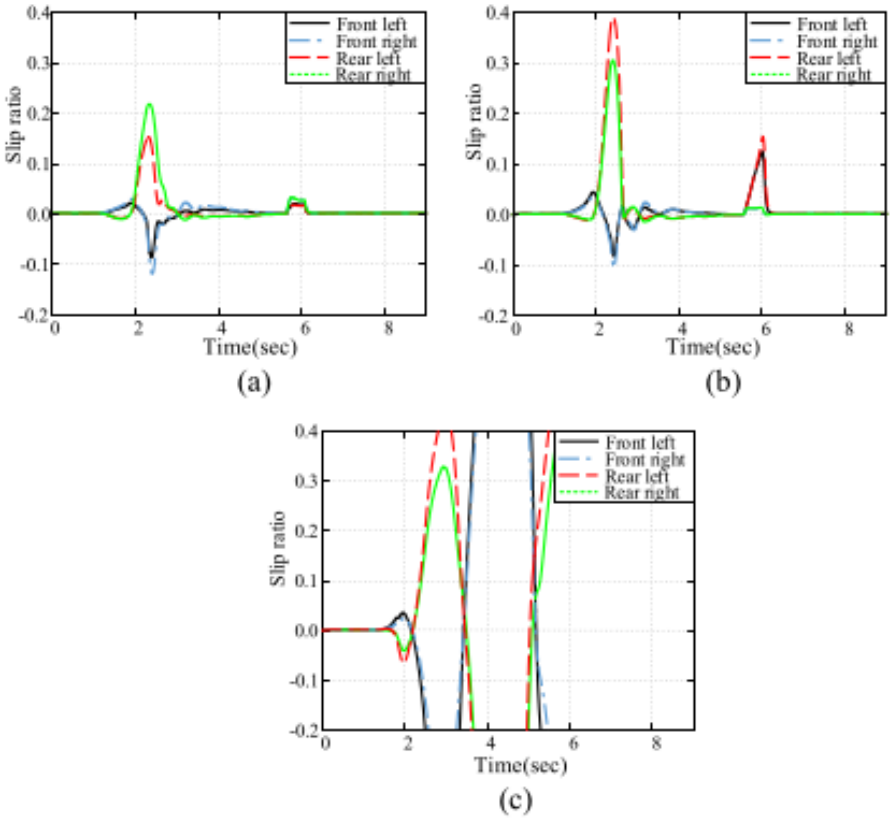

Fig. 14. Slip ratio: (a) holistic controller, (b) hierarchical controller, (c) sepa- rate controller [1]

structure also produces slip ratios that are oversized beyond the normal ranges whereas the holistic controller maintains its slip ratios in the normal range and hence the proposed controller is very much advantageous that the previous controllers.

We also tried to simulate the same results using MATLAB. Since we do not have access to the Carsim Simulation platform, some results involving vehicle dynamics could not be simulated.

TABLE IV

ADDITIONAL PARAMETERS USED IN CODE

Symbol	Definition	Value
N_p	No. of Steps in Prediction Horizon	200 steps
N_c	No. of Steps in Control Horizon	40 steps
r_w	Radius of Wheel	0.35179 m
m_w	Mass of Wheel	24.5 kg
$W_{e_{x0}}$	Origin weight of e_x	0.01
$W_{e_{yp}}$	Weight of e_{yp}	0.01
$W_{\Delta\psi}$	Weight of $\Delta\psi$	0.01
W_{v_y}	Weight of v_y	0.01
W_{δ_f}	Weight of steering angle	0.0001
W_{Fij0}	Origin Weight of longitudinal forces on tire	0.0001
$Y_desired$	Desired Output Vector of size $4N_p \times 1$	Vector of zeros
ρ	Weight of slack variable	0.0001
ε_0	Initial value of Slack Variable	0.01
ε_max	Maximum Value of Slack variable	0.1
a_s	Parameter in Weight Adaptive Scheme	0.001
b_s	Parameter in Weight Adaptive Scheme	-0.0008
c_s	Parameter in Weight Adaptive Scheme	20
k_s	Parameter in Weight Adaptive Scheme	20
k_w	Parameter in Weight Adaptive Scheme	20
e_{ythe}	Parameter in Weight Adaptive Scheme	0.5 m
$\Delta\psi_{the}$	Parameter in Weight Adaptive Scheme	5 rad
β_{the}	Parameter in Weight Adaptive Scheme	$\frac{15\pi}{180}$ rad
u_{min}	Minimum value of control input	$[-0.7\ 0\ 0\ 0\ 0]^T$
u_{max}	Maximum value of control input	$[0.7\ \frac{\mu mg}{4}\ \frac{\mu mg}{4}\ \frac{\mu mg}{4}\ \frac{\mu mg}{4}]^T$
Δu_{max}	Maximum change in control input per time step	$[0.03\ 0.43\frac{\mu mg}{4}\ 0.43\frac{\mu mg}{4}\ 0.43\frac{\mu mg}{4}\ 0.43\frac{\mu mg}{4}]^T$
y_{min}	Minimum value of Output	$[-5\ -5\ -0.1\ -5]^T$
y_{max}	Minimum value of Output	$[5\ 5\ 0.1\ 5]^T$

Below is the code of the control algorithm that we tried to code in MATLAB:

```
cf = 117000;
cr = 108000;
lf = 1.4;
lr = 1.65;
Iz = 3234;
```

```
m = 1650; %mass of the vehicle
rw = 0.35179;
vxd = 25;
vx = vxd; %vehicle velocity

delta_t = 0.4; % preview time
W = 1.861;
dT = 0.01; % Time step
Np = 2.0/dT; % Prediction Horizon
Nc = 0.2*Np;
nu = 0.7; % Friction Coeff
m_w = 24.5; % mass of wheel
I_w = m_w*rw*rw/2; % Inertia of wheel
g = 9.81;

C = [1 0 0 0 0; 0 1 0 0 0; 0 0 1 0 0 ; 0 0 0 1 0];
Cd = C;
X = zeros(5*Np, 1);
Y = zeros(4*Np, 1);
Y_des = zeros(4*Np, 1);
U = zeros(5*Nc, 1);



% disp(size(Dp))
Q = 0.01*eye([4*Np, 4*Np]);
R = 0.0001*eye([5*Nc, 5*Nc]);

Ydes = [0; 0; 0; 0];
rho = 0.0001;
epsilon = 0.01;
```

```
epsilon_max = 0.1;
k_curv = 0;
t = 0;
x = zeros(5, 10/dT);
Gk = zeros(5*Nc+1, 5*Nc+1);
Hk = zeros(1, 5*Nc+1);
Wex0 = 0.01;
Wf0 = 0.0001;
Wey = 0.01;
Wdpsi = 0.01;
Wvy = 0.01;
as = 0.001;
bs = -0.0008;
cs = 20;
ks = 20;
kw = 20;
Kr = zeros(4,1);
Wf = zeros(4,1);
ey_the = 0.5;
dpsi_the = 5;
beta_the = 15*pi/180;
umin = [-0.7; 0; 0; 0; 0];
umax = [0.7; 0.25*(nu*m*g + m*1); 0.25*(nu*m*g + m*1); 0.25*(nu*m
   *g + m*1); 0.25*(nu*m*g + m*1)];
dumax = [0.03; 0.43*0.25*(nu*m*g + m*1); 0.43*0.25*(nu*m*g + m*1)
   ; 0.43*0.25*(nu*m*g + m*1); 0.43*0.25*(nu*m*g + m*1)];
ymin = [-5; -5; -0.1; -5];
ymax = [5; 5; 0.1; 5];

up = [0 ; nu*m*g/4; nu*m*g/4; nu*m*g/4; nu*m*g/4];
Umax = umax;
```

```
Umin = umin;
Ymin = ymin;
Ymax = ymax;
y = zeros(4, 10/dT);
y(:,1) = Cd*x(:,1);
for i = 2:Nc
    Umax = cat(1,Umax, umax);
    Umin = cat(1,Umin, umin);
end
Umax = cat(1,Umax, epsilon_max);
Umin = cat(1,Umin, 0);

for i=2:Np
    Ydes = cat(1, Ydes, [0; 0; 0; 0]);
    Ymax = cat(1,Ymax,ymax);
    Ymin = cat(1, Ymin, ymin);
end
for k = 1:1:((10/dT) - 1)


    if t<2
       k_curv = 0;
    elseif t<3
        k_curv = 0.01*(t-2);
    elseif t<5
        k_curv = 0.01;
    elseif t<6
        k_curv = -0.01*t + 0.06;
    else
        k_curv = 0;
    end
```

```
t = t + dT;
ex = y(1,k);
eyp = y(2,k);
dpsi = y(3,k);
vy = y(4,k);
vx = ex + vxd;
beta = atan2(vy, vx);
DL = vx * delta_t; % preview distance
w = [(4*vx)/(rw); (4*vx)/(rw); (4*vx)/(rw); (4*vx)/(rw)]; %
   angular wheel speed
A = [0 0 0 0 0 ; 0 0 vx 1 DL ; 0 0 0 0 1; 0 0 0 -((cf+cr)/(m*
   vx)) -vx - (cf*lf - cr*lr)/(m*vx); 0 0 0 (cr*lr -cf*lf)/(
   Iz*vx) -(cf*lf*lf + cr*lr*lr)/(Iz*vx)];

B = [0 1/m 1/m 1/m 1/m ; 0 0 vx 1 DL; 0 0 0 0 1; cf/m 0 0 0
   0; (cf*lf)/Iz -W/(2*Iz) W/(2*Iz) -W/(2*Iz) W/(2*Iz)];
Ad = eye([5,5]) + A*dT;
Bd = B*dT;
Cp = Cd*Ad;
for i=2:1:Np
    Cp = cat(1, Cp, Cd*(Adˆi));
end
%disp(size(Cp));

Ep = cat(2, Cd, zeros(4,5*(Np - 1)));

for i = 2:Np
    Ep_row = Cd*Adˆ(i-1);
    for j = 2:Np
        if j >i
            Ep_row = cat(2, Ep_row, zeros(4,5));
```

```
        elseif j == i
            Ep_row = cat(2, Ep_row, Cd);

        else
            Ep_row = cat(2, Ep_row, Cd*Adˆ(i-j));
        end
    end
    Ep = cat(1, Ep, Ep_row);
end

%disp(Ep(9:12, 1:15));

Dp = cat(2, Cd*Bd, zeros(4, 5*(Nc - 1)));
for i = 2:Np
    Dp_row = Cd*(Adˆ(i-1));
    for j = 2:Nc
        if i <= Nc
            if j >i
                Dp_row = cat(2, Dp_row, zeros(4,5));
            elseif j == i
                Dp_row = cat(2, Dp_row, Cd*Bd);
            else
                Dp_row = cat(2, Dp_row, Cd*(Adˆ(i-j)));
            end
        else
            Dp_row = cat(2, Dp_row, Cd*(Adˆ(i-j)));
        end
    end
    Dp = cat(1, Dp, Dp_row);
end
```

```
Acons = cat(1,Dp, -Dp);
Acons = cat(2, Acons, -1*ones(8*Np, 1));


Qs = max(abs(eyp)/ey_the, max(abs(dpsi)/dpsi_the, abs(beta)/
   beta_the));
if k ~= 1
    w(1) = w(1) + up(2)*rw*dT/(I_w*cos(up(1)*pi/180)); %
       Angular Wheel Speeds
    w(2) = w(2) + up(3)*rw*dT/(I_w*cos(up(1)*pi/180));
    w(3) = w(3) + up(4)*rw*dT/I_w;
    w(4) = w(4) + up(5)*rw*dT/I_w;
end

for z = 1:4
    if vx > rw*w(z)
        Kr(z) = abs(rw*w(z)/vx - 1);
    else
        Kr(z) = abs(1 - vx/(rw*w(z)));
    end
    if(Kr(z) <= 0.1)
        Wf(z) = Wf0;
    else
        Wf(z) = Wf0*exp(kw*(Kr(z) -0.1));
    end
end

for j = 1:Np
    if Qs <=1
        Q(4*j-3,4*j-3) = Wex0;
    else
```

```
            Q(4*j-3,4*j-3) = as*bs*tanh((ks/Qs)^cs);
        end
    end

    for o = 1:Nc
        R(5*o - 3, 5*o - 3) = Wf(1);
        R(5*o - 2, 5*o - 2) = Wf(2);
        R(5*o - 1, 5*o - 1) = Wf(3);
        R(5*o, 5*o) = Wf(4);
    end
    Gk(1:(5*Nc), 1:(5*Nc)) = 2*(transpose(Dp)*Q*Dp + R);
    Gk(5*Nc+1, 5*Nc+1) = 2*rho;
    Hk(1:5*Nc) = 2*transpose(Cp*x(:,k)-Ydes)*Q*Dp;
    Umin(1:5) = max(umin, up-dumax);
    Umax(1:5) = min(umax, up+dumax);
    Bcons = Ymax - Cp*x(:,k);
    Bcons = cat(1,Bcons, -Ymin+Cp*x(:,k));
    disp(size(Gk));
    U = quadprog(Gk, transpose(Hk), Acons, Bcons, [], [], [],
       Umax);
    disp(size(U));
    up = U(1:5);

    w_dist = [(vxd - vx)/(Np*dT); vx*k_curv; 0; 0; 0];

    x(:,k+1) = Ad*x(:,k) + Bd*U(1:5) + dT*w_dist;
    y(:, k+1) = Cd*x(:, k+1);
end
```

On simulating in MATLAB using quadprog() for Case one, considering no lower bound for U ($U_{min} = [\,]$) and U_{max} based on parameters shown in Table IV, we get the results as shown in Fig. 15. In Fig. 15, the X-axis denotes the No. of steps and Y-axis denotes the output variable.

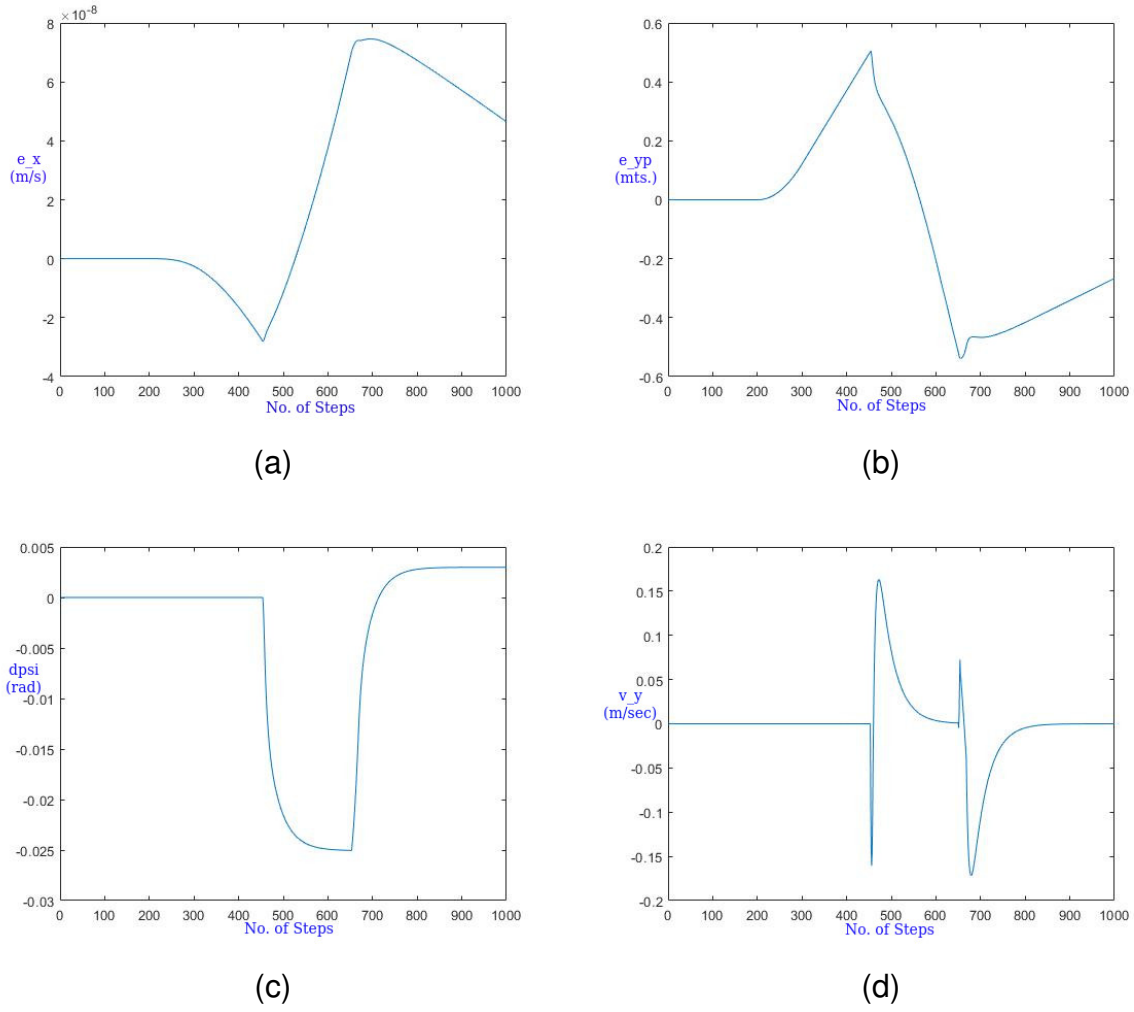

Fig. 15. Simulation Results of (a) e_x, (b) e_{yp}, (c) $\Delta\psi$, (d) v_y

We can see that the errors are approaching 0 or a small enough number.

The code contains a lot of unknown parameters and weights (Table IV) which need to be fine tuned for this particular model as the performance of the proposed model depends largely on the weights that are used in the weight adaptive mechanism. Since we were not able to fine tune these parameters the results that we are getting are not exact results but pretty close to results mentioned in the paper. We also tried looking for some research papers with similar approaches like [6] and [7] to get an approximate value of these parameters and weights and some of the parameters in the code and Table IV have been taken from these papers. Finally, we also mailed the author of this paper but he has not replied yet.

V. Conclusion

The novel holistic adaptive multi-model predictive path following controller for 4WID autonomous vehicle proposed in this paper has a linear structure since the complex relationships are decoupled which increases the real time performance of the MPC. The weight adaptive mechanism ensures that the controller handles extreme conditions as mentioned in case two of simulation. The simulation results also show that the proposed controller achieves excellent performance in both normal and extreme driving conditions. In normal conditions the multiple-model adaptive law effectively improves the performance while in extreme driving conditions, the weighting adaptive mechanism helps maintain the road safety and the dynamics stability. This concludes that the proposed model is superior over other control methods that are used in 4WID autonomous vehicles.

References

[1] Y. Liang, Y. Li, A. Khajepour and L. Zheng, "Holistic Adaptive Multi-Model Predictive Control for the Path Following of 4WID Autonomous Vehicles," in IEEE Transactions on Vehicular Technology, vol. 70, no. 1, pp. 69-81, Jan. 2021, doi: 10.1109/TVT.2020.3046052.

[2] H. Zengin, N. Zengin, B. Fidan, and A. Khajepour, "Blending based multiple-model adaptive control for multivariable systems and application to lateral vehicle dynamics," in Proc. 18th Eur. Control Conf., Naples, Italy, 2019, pp. 2957–2962.

[3] P. Ioannou and B. Fidan, "Adaptive control tutorial," Soc. for Ind. Appl. Math., Philadelphia, United States, vol. 11, 2006.

[4] L. Yuan et al., "Nonlinear MPC-based slip control for electric vehicles with vehicle safety constraints," Mechatronics, vol. 38, pp. 1–15, Sep. 2016.

[5] H. Zengin, "Multiple-model robust adaptive vehicle motion control", Ph.D. dissertation, Univ. Waterloo, Waterloo, Canada, 2019.

[6] M. I. Palmqvist, "Model Predictive Control for Autonomous Driving of a Truck", Degree Project, Electrical Engineering Dept., KTH Royal Institute of Technology, Stockholm, Sweden, 2016.

[7] G. Bai, Y. Meng, L. Liu, W. Luo, Q. Gu, and K. Li, "A New Path Tracking Method Based on Multilayer Model Predictive Control," Applied Sciences, vol. 9, no. 13, p. 2649, Jun. 2019.