Model Predictive Control for Integrated Lateral Stability

Jad Yahya
ID: 30385616189

Brandon Chin
ID: 3036183435

C.K. Wolfe
ID: 3037461544

Junjie Chen
ID: 3035980492

***Abstract*—This paper addresses the design of a Model Predictive Controller (MPC) for integrated lateral stability, traction/braking control, and rollover prevention of electric vehicles. The dynamic model used in this paper is superior to the literature in that it includes rollover prevention into the MPC (resulting in a total of 8 states) but linearizes the tire model prior to solving the MPC problem to save computation time. Building on that, the design of a model predictive controller for lateral stability control of an autonomous racing car for the Indy Autonomous Challenge is proposed. The latter includes the road banking angle, lateral position tracking, and a different suspension model.**
YouTube Video Link

***Index Terms*—model predictive control, advanced driver assistance systems, controller design**

I. Introduction

Active stability systems for vehicle control and driver assistance have become excessively used in applications during the past decades[1][2][3]. In the sense of lateral stability, the role of active safety control systems is to make sure the vehicle remains in the stable handling region in the face of adversity, such as on slippery surfaces or under aggressive maneuvers, within the laws of physics. On the contrary, the longitudinal stability control system ensures that the vehicle wheel spin is regulated so that the braking/acceleration torque is used optimally. Moreover, the rollover control system ensures untripped rollovers do not occur by regulating lateral load transfer from the inner to the outer wheels.

Model predictive control (MPC) has become more extensively employed for vehicle stability control [1][2][3] due to its undeniable advantages. These include, but are not limited to, being able to predict over a finite horizon, being able to deal with multiple input multiple output (MIMO) systems, and being able to deal with constraints. Multiple works in literature have considered nonlinear MPC models (NMPC) for the purposes stated earlier, which results in high computational time [2][4]. In [2], the NMPC choice results in simplifying the model to only control lateral dynamics without accounting for the coupling effects due to load transfer. In [4], the model is reduced to in-plane motion using a bicycle vehicle model. Hence, an NMPC results in the requirement to compromise the extent of the control by decreasing the number of states controlled.

The main objective of this paper is to recreate the results in [1] which aim to develop a linear model predictive controller for slip control, handling improvement, lateral stability control, and rollover prevention of electric vehicles. This is achieved by linearizing the selected vehicle tire model at each solver iteration. Namely, we will be linearizing a Pacejka and a Dugoff tire model (only the Dugoff model will be considered for performance assessment). In addition to that, the study develops a centralized MPC for lateral stability and lateral trajectory tracking of an autonomous racing car while accounting for the effects of a load shift. This controller is intended to be used in the Dallara AV-21 vehicle for AI Racing Tech in the Indy Autonomous Challenge.

II. Vehicle Dynamic Model

The dynamic model that is used in this paper for both the general electric vehicle and the Dallara AV-21 is an in-plane model with roll dynamics. To develop this model, the tire model is initially linearized about an equilibrium point into the following form:

$$\begin{aligned} f_{yi} &= \bar{f}_{yi} + \tilde{c}_{\alpha_i}\left(\alpha_i - \bar{\alpha}_i\right) \quad \text{for } i = 1 \text{ to } 4 \\ f_{xi} &= \frac{Q_i}{R_w} \end{aligned} \tag{1}$$

where f_{yi} is the tire force, $\bar{f}_{yi}$ and $\tilde{c}_{\alpha_i}$ represent the lateral tire force and the cornering coefficient at the side slip angle of the operating point, respectively, and the front-left, front-right, rear-left, and rear-right wheels are labeled as wheel number 1–4, respectively. f_{xi} is the longitudinal force on the tires, Q_i is the torque on each wheel and R_w is the wheel's effective radius. But, from vehicle dynamics, we know that:

$$\alpha_i = \delta_i - \frac{v + a_i r}{u} \tag{2}$$

We note the following nomenclature that will be used: δ_i is the steering angle for tire i, u, v, and r are the longitudinal velocity, lateral velocity, and yaw rate and φ is the roll angle. Also, l_f and l_r are the horizontal distances of the CG to the front and rear axles, respectively. We define:

$$a_i = \begin{cases} l_f & i = 1, 2 \\ -l_r & i = 3, 4 \end{cases} \tag{3}$$

Using equations 1, 2, and 3 we can develop the following equation

$$f_i = B_{1i}X_b + B_{2i}W_i + D_{1i} \tag{4}$$

where:

$$f_i = [f_{xi}, f_{yi}]^T \tag{5}$$

$$W_i = [Q_i, \delta_i]^T \tag{6}$$

$$B_{1i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{\tilde{\alpha}_{\alpha_i}}{u} & -\frac{a_i\tilde{c}_{\alpha_i}}{u} & 0 & 0 \end{bmatrix}, B_{2i} = \begin{bmatrix} \frac{1}{R_w} & 0 \\ 0 & \tilde{c}_{\alpha_i} \end{bmatrix} \tag{7}$$

$$D_{1i} = \begin{bmatrix} 0 \\ \bar{f}_{yi} - \tilde{c}_{\alpha_i}\bar{\alpha}_i \end{bmatrix} \tag{8}$$

$$X_b = [v, r, \varphi, \dot{\varphi}]^T \tag{9}$$

We further define a local actuator reconfiguration matrix for each wheel. It is a diagonal matrix that takes binary entries for its diagonal entries to indicate whether the actuator is available for vehicle control or not and is given by:

$$T_{wi} = \begin{bmatrix} t_{Qi} & 0 \\ 0 & t_{\delta i} \end{bmatrix} \tag{10}$$

In addition, a mapping matrix L_{wi} is defined for the mapping from the local tire forces to vehicle corner forces (in the direction of the axes attached to the vehicle's center of gravity).

$$L_{wi} = \begin{bmatrix} \cos\delta_i & -\sin\delta_i \\ \sin\delta_i & \cos\delta_i \end{bmatrix} \tag{11}$$

the corner forces at the i_{th} wheel is defines as

$$F_{ci} = [F_{xi}, F_{yi}]^T \tag{12}$$

The control input utilized is given by

$$\begin{aligned} \delta f_i &= B_{2i}T_{wi}U_i \\ \delta f_i &= [\delta f_{xi}, \delta f_{yi}]^T \\ U_i &= [\Delta Q_i, \Delta\delta_i]^T \end{aligned} \tag{13}$$

Utilizing equations 4, 10,11,12, and 13, the following equation is developed:

$$F_{ci} = L_{wi}\left(B_{1i}X_b + B_{2i}W_i + D_{1i} + B_{2i}T_{wi}U_i\right) \tag{14}$$

Combining the equations for all 4 tires into one equation yields:

$$F_c = L_w\left(B_1X_b + B_2W + D_1 + B_2T_wU\right) \tag{15}$$

where:

$$\begin{aligned} F_c &= \left[F_{c1}^T F_{c2}^T F_{c3}^T F_{c4}^T\right]^T \\ U &= \left[U_1^T U_2^T U_3^T U_4^T\right]^T \\ W &= \left[W_1^T W_2^T W_3^T W_4^T\right]^T \\ L_w &= \text{blockdiag}\left(L_{w1}, L_{w2}, L_{w3}, L_{w4}\right) \\ B_1 &= \left[B_{11}^T B_{12}^T B_{13}^T B_{14}^T\right]^T \\ B_2 &= \text{blockdiag}\left(B_{21}, B_{22}, B_{23}, B_{24}\right) \\ D_1 &= \left[D_{11}^T D_{12}^T D_{13}^T D_{14}^T\right]^T \\ T_w &= \text{blockdiag}\left(T_{w1}, T_{w2}, T_{w3}, T_{w4}\right) \end{aligned}$$

Moreover, the corner forces are translated into forces at the vehicle's center of gravity. The force vector on the center of gravity, denoted by F, can be expressed as:

$$F = \begin{bmatrix} F_X & F_Y & M_Z \end{bmatrix}^T \tag{16}$$

Define L_c the mapping matrix from corner forces to CG forces as

$$L_c = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ -t_f/2 & l_f & t_f/2 & l_f & -t_r/2 & -l_r & t_r/2 & -l_r \end{bmatrix} \tag{17}$$

where t_f and t_r are the vehicle's front and rear track widths. Thus, the following relation is established

$$F = L_cF_c \tag{18}$$

Fig. 1 below shows the different kinds of forces on the vehicle. In blue are the tire forces, in red are the corner forces, and in the direction of the red vectors are the forces on the vehicle's center of gravity. From here on, the differences

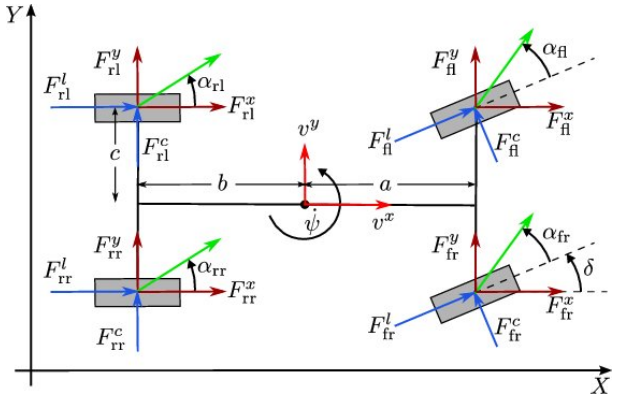

Fig. 1. Diagram Visualizing the Different Kinds of Forces

between the general electric vehicle model and the Dallara AV-21 model become pronounced. The different models will be discussed in the below subsections.

A. General Electric Vehicle

The vehicle body dynamics for the general electric vehicle are derived from Newton's equations of motion.

$$\dot{X}_b = A_FX_b + B_FF \tag{19}$$

where

$$A_F = \begin{bmatrix} 0 & -u & \frac{m_sh_s(k_\varphi - m_sgh_s)}{(-m_s^2h_s^2 + mI_{xx})} & \frac{m_sh_sc_\varphi}{(-m_s^2h_s^2 + mI_{xx})} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-m(k_\varphi - m_sgh_s)}{(-m_s^2h_s^2 + mI_{xx})} & \frac{-mc_\varphi}{(-m_s^2h_s^2 + mI_{xx})} \end{bmatrix} \tag{20}$$

$$B_F = \begin{bmatrix} 0 & \frac{I_{xx}}{(-m_s^2h_s^2 + mI_{xx})} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 \\ 0 & \frac{m_sh_s}{(-m_s^2h_s^2 + mI_{xx})} & 0 \end{bmatrix} \tag{21}$$

where m and m_s are the total mass and the sprung mass, respectively. I_{xx} and I_{zz} are the roll and yaw moments of inertia, g is the gravitational acceleration, h_s is the distance between the vehicle's center of gravity and its roll center, and k_φ and c_φ represent the effective torsional stiffness and torsional damping in the roll direction, respectively. Joining equation 18 and equation 19 yields

$$\begin{aligned}\dot{X}_b = &\left(A_F + B_F L_c L_w B_1\right) X_b + \left(B_F L_c L_w B_2\right) W \\ &+ \left(B_F L_c L_w B_2 T_w\right) U + B_F L_c L_w D_1\end{aligned} \tag{22}$$

And the wheel dynamics are derived using

$$I_w \dot{\omega}_i = Q_i + \Delta Q_i - R_w f_{xi} \tag{23}$$

where I_w and ω_i are the rotational moment of inertia and the rotational speed. Defining $X_w = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T$, the state-space equation for all four wheels can be written as

$$\dot{X}_w = A_w X_w + E_w W + B_w U + D_w \tag{24}$$

Finally, the full standard state-space form is expressed as

$$\begin{aligned}
&\dot{X} = AX + EW + BU + D \\
&X = [v, r, \varphi, \dot{\varphi}, \omega_1, \omega_2, \omega_3, \omega_4]^T \\
&U = [\Delta Q_1 \Delta\delta_1 \Delta Q_2 \Delta\delta_2 \Delta Q_3 \Delta\delta_3 \Delta Q_4 \Delta\delta_4]^T \\
&A = \text{blockdiag}\left(A_b, A_w\right) \\
&E = \left[E_b^T E_w^T\right]^T \\
&B = \begin{bmatrix} B_b^T & B_w^T \end{bmatrix}^T \\
&D = \left[D_b^T D_w^T\right]^T \\
&A_b = A_F + B_F L_c L_w B_1 \\
&E_b = B_F L_c L_w B_2 \\
&B_b = B_F L_c L_w B_2 T_w \\
&D_b = B_F L_c L_w D_1
\end{aligned} \tag{25}$$

B. Dallara AV-21 for Indy Autonomous Challenge

The vehicle dynamics in the case of the Dallara AV-21 have been modified, albeit still derived from Newton's equations of motion, and are shown below.

$$\dot{X}_b = A_F X_b + B_F F + C_\phi \phi_r \tag{26}$$

where

$$A_F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -u & \frac{2gI_{xx}m + 2gh_s^2 m^2 + h_s k_s l_s^2 m_s}{2m(I_{xx} + h_s^2 m + h_s^2 m_s)} & \frac{b_s h_s l_s^2 m_s}{(2m(I_{xx} + h_s^2 m + h_s^2 m_s))} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{2gh_s m^2 - k_s * l_s^2 m}{2m(I_{xx} + h_s^2 m + h_s^2 m_s)} & -\frac{b_s l_s^2 m}{2m(I_{xx} + h_s^2 m + h_s^2 m_s)} \end{bmatrix}$$

$$B_F = \begin{bmatrix} 0 & 0 \\ 0 & \frac{I_x x + m h_s^2}{2m(I_{XX} + h_s^2 m + h_s^2 * m_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{2m_s h_s}{2m(I_{XX} + h_s^2 m + h_s^2 * m_s} \end{bmatrix} \quad C_\phi = \begin{bmatrix} 0 \\ \frac{2gm_s m h_s^2 - h_s k_s l_s^2 m_s}{2m(I_{XX} + h_s^2 m + h_s^2 * m_s} \\ 0 \\ 0 \\ \frac{k_s l_s^2 m - 2gh_s m^2}{2m(I_{XX} + h_s^2 m + h_s^2 * m_s} \end{bmatrix}$$

$$0 \quad 0 \quad \frac{1}{I_z} \quad 0 \quad 0 \tag{27}$$

$$X_b = \begin{bmatrix} y & v_y & r & \varphi & \dot{\varphi} \end{bmatrix}^T$$

where φ_r is the road banking angle, k_s is the spring stiffness and b_s is the damping of the damper.

Additionally, in this case, the changes implemented are adding the lateral position as a state in order to ensure the racing line is tracked for the autonomous racing car. Moreover, the banking angle is added to the model since oval tracks tend to have significant banking angles. This entailed a small angle assumption on $\varphi - \phi_r$ which is common for vehicle dynamics applications. Moreover, the suspension model was changed from including the entire vehicle to a spring-damper model. We are aware that the stiffness and damping values are different for the front and rear suspension, but for the purposes of this application we test the effect of equating them.

Lastly, we do not consider tire longitudinal slip control and hence do not include the wheel rotational speeds as states. The reasoning behind this is that the vehicles used in the Indy Autonomous Challenge do not hit their maximum acceleration and hence should not need this kind of control (especially at the cost of adding 4 states to the MPC). Moreover, in the case that is to happen, the team has already developed an ABS system to ensure no excessive tire slip occurs. Hence, the state space model in this case is given by

$$\begin{aligned}\dot{X}_b = &\left(A_F + B_F L_c L_w B_1\right) X_b + \left(B_F L_c L_w B_2\right) W \\ &+ \left(B_F L_c L_w B_2 T_w\right) U + B_F L_c L_w D_1 + C_\phi \phi_r\end{aligned} \tag{28}$$

III. MPC Design and Algorithm

In formulating the MPC problem, we use two different approaches for the different cases. This will be elaborated on in the following subsections.

A. General Electric Vehicle

After developing the vehicle dynamic model, a controller is recreated from the work in [1] that considers integrated handling improvement, vehicle lateral stability, rollover prevention, and slip control in braking and traction for electric vehicles. First, we consider yaw rate tracking which requires a desired yaw rate. That is considered to be the response of a linear bicycle model given by

$$r_b = \frac{u}{l + k_{us\_d} u^2} \delta_d \tag{29}$$

where l is the vehicle's wheelbase, δ_d is the driver's steering angle,and k_{us_d} is the desired under-steer coefficient for the vehicle. However, by employing the maximum lateral force on each axle, we get the below relation

$$r_{\max} = \frac{\mu_y g}{u} \tag{30}$$

where μ_y is the lateral friction coefficient. So, the desired yaw rate is defined by

$$r_d = \text{sign}\left(\delta_d\right) \times \min\left(\left|r_b\right|, r_{\max}\right) \tag{31}$$

Next, we consider the lateral stability constraint. To do so, we limit the side-slip angle of the rear tire as

$$\left| \frac{-v_y}{u} + \frac{l_r}{u} r \right| < \alpha_{r\_\max} \tag{32}$$

where $\alpha_{r\_\max}$ is the maximum allowable side-slip angle for the rear tire. Third, we resort to the rollover index for rollover prevention. Namely

$$RI = C_1\varphi + C_2\dot{\varphi} \tag{33}$$

where

$$\begin{aligned} C_1 &= \frac{2}{mgT}\left(k_\varphi\left(1+\frac{m_s h_R + m_u h_u}{m_s h_s}\right) - (m_s h_R + m_u h_u)\, g\right) \\ C_2 &= \frac{2c_\varphi}{mgT}\left(1+\frac{m_s h_R + m_u h_u}{m_s h_s}\right) \end{aligned} \tag{34}$$

where m_u is the unsprung mass, h_u is the CG height of the unsprung mass, and h_R is the distance of the roll center to the ground.
Finally, for slip control for traction and braking, we resort to the following expression [1]

$$\omega_{i-c} = \frac{u}{R_w} \pm \lambda_{\max} \max\left(\frac{u}{R_w}, \omega_i\right) \tag{35}$$

where λ_{max} is the maximum allowable slip ratio.The set of state constraints turns out to be

$$\begin{aligned} -RI_{max} \le RI \le RI_{max} \\ -r_{max} \le r \le r_{max} \\ min\{\omega_{i-c}\} \le \omega_i \le max\{\omega_{i-c}\} \\ -\alpha_{r\_\max} \le \frac{v_y}{u} + \frac{l_r}{u} r \le \alpha_{r\_\max} \end{aligned} \tag{36}$$

As for actuator constraints, the ones considered in this problem relate to the maximum possible torque generated, the maximum tire force capacity according to the friction between tires and the road, and the maximum steering angle.

$$\begin{aligned} Q_i^{min} - Q_i(0) \le \Delta Q_i^k \le Q_i^{max} - Q_i(0) \\ -f_{xi}^p(0) - Q_i(0) \le \Delta Q_i^k \le f_{xi}^P(0) - Q_i(0) \\ \delta_i^{min} - \delta_i(0) \le \Delta\delta_i^k \le \delta_i^{max} - \delta_i(0) \end{aligned} \tag{37}$$

where $Q_i(0)$ is the driver command at the beginning of the corresponding horizon, $\delta_i(0)$ is the driver steering command at the beginning of the corresponding horizon, and $f_{xi}^p(0) = \mu_x f_{zi}(0)\sqrt{1-\left(\frac{f_{yi}(0)}{\mu_y f_{zi}(0)}\right)^2}$. Also, $f_{zi}(0)$ and $f_{yi}(0)$ are the vertical and lateral forces at the beginning of the horizon.

With these constraints defined as shown in equations 33 and 34, the constrained finite time optimal control problem (CFTOC) is formulated in Pyomo to track a desired state vector given by

$$\bar{X_d} = \begin{bmatrix} 0 & r_d & 0 & 0 & \frac{u}{R_w} & \frac{u}{R_w} & \frac{u}{R_w} & \frac{u}{R_w} \end{bmatrix} \tag{38}$$

B. Dallara AV-21 for Indy Autonomous Challenge

The MPC design in this case should maintain lateral stability and lateral race line tracking. Hence, it would use the same actuator constraints as those in equation 37. However, from equation 36, we only need to consider the second and fourth equations. Moreover, here the front steering angles should be equal and the rear wheel torques should be equal since that is the case for the Dallara AV-21. Also, the front wheels aren't driven and the rear wheels cannot steer. This last point is reflected in the choice of entries in equation 10. Following identifying the constraints, we know we want to track the following set of states

$$\bar{X}_d = [y_d, 0, r_d, 0, 0] \tag{39}$$

The problem was formulated as a CFTOC in Pyomo. Moreover, to make this feasible for real-time implementation, we have been working on Developing it as a 2-Norm Quadratic Program with substitution in CasADi and solving it using the solver 'qpoases'.
The desired steering angle for tracking the racing line is developed by utilizing checkpoints provided by a path planner. The latter provides a desired coordinate to track (x_d, y_d). This can then be utilized to determine the desired heading angle as $\psi_d = \text{atan}(\frac{y_d}{x_d})$. Therefore, the desired steering angle is determined as $\delta_d = \psi_d + \frac{v_y}{u}$.

IV. Parameters

A. General Electric Vehicle

While we are actively working as part of our capstone project on developing a high-fidelity simulator through using a DLL from one of our sponsors (ChassisSim), for the purposes of this project we primarily relied on a model with full-state feedback and perfect model knowledge. For a general electric vehicle, the parameters used are defined in the below table.

TABLE I
Vehicle Parameters for General Electric Vehicle

Parameter	Value	Parameter	Value
$m_s(kg)$	1590	$m_u(kg)$	270
$t_f = t_r(m)$	1.575	$h_c g(m)$	0.72
$l_f(m)$	1.18	$l_r(m)$	1.77
$I_{xx}(kgm^2)$	894.4	$I_{zz}(kgm^2)$	2687.1
$h_s(m)$	0.57	$h_u(m)$	0.2
$k_\varphi(Nm/rad)$	189506	$c_\varphi(Nm/rads)$	6364
$r_{eff}(m)$	0.393	$I_w(kgm^2)$	1.1
μ_x	1	μ_y	1
$Q_{max}(Nm)$	1600	$Q_{min}(Nm)$	-1600
$\delta_{max}(rad)$	1	RI_c	0.7
$C_\sigma(N)$	80000	$C_\alpha(N/rad)$	47275
$k_{usd}(rads/m^2)$	0.4		

The MPC parameters are given in table II. Additionally, it is assumed that at the start of the horizon the weight is evenly distributed at the four vehicle corners. Moreover, the lateral forces on the tires are zero (i.e, the vehicle is moving in a straight line). Initially, the torques on the rear wheels are $100Nm$ and the other actuators aren't used. Ultimately, the MPC would use torque vectoring to ensure the vehicle is laterally stable while tracking the driver's desired yaw rate.

TABLE II
MPC PARAMETERS FOR GENERAL ELECTRIC VEHICLE

Parameter	Value	Parameter	Value
N	10	M	250
$u(kph)$	80	$T_s(s)$	0.1
$x_0(SI)$	$[0,0,0,0,53,53,53,53]^T$	σ	0.1
$\alpha_{max}(deg)$	6	$\delta_d(rad)$	0.15

B. Dallara AV-21 for Indy Autonomous Challenge

For the Dallara AV-21, we know the following most of the parameters defined in table III. However, the cornering and longitudinal tire stiffness were estimated through consulting with Danny Nolan (Founder of ChassisSim). Moreover, the friction coefficient values have been set to 1, but in reality, they would need to be estimated in real-time.

TABLE III
VEHICLE PARAMETERS FOR DALLARA AV-21

y

Parameter	Value	Parameter	Value
$r_{eff}(m)$	0.29	$l_f(m)$	1.6566
$l_r(m)$	1.3152	$m(kg)$	803.182
$m_s(kg)$	672.2	$m_f(kg)$	355.45
$I_{xx}(kgm^2)$	200	$h_s(m)$	0.1
$I_z(kgm^2)$	1200	$C_\sigma(N)$	10000
$C_\alpha(N/rad)$	8000	μ_x	1
μ_y	1	$K_s(N/m)$	10000
$b_s(N/ms)$	1000	$\delta_{max}(rad)$	0.1

TABLE IV
MPC PARAMETERS FOR DALLARA AV-21

Parameter	Value	Parameter	Value
N	50	M	140
$u(kph)$	180	$T_s(s)$	0.05
$x_0(SI)$	$[0,0,0,0,0]^T$	$\alpha_{max}(deg)$	6

V. PERFORMANCE

A. General Electric Vehicle

Under the conditions shown in table IV, our results for the simulation of the response of an ideal vehicle with our MPC controller implemented is shown in figures 2, and 3. The actuator configuration matrix is selected such that the vehicle is controlled purely by torque vectoring. We can see clearly that the steady state yaw rate is different from the commanded one by the driver. The reason for this is that this is the maximum attainable yaw rate that ensures the constraints are satisfied and the vehicle is stable i.e., the driver is safe. Moreover, figure 3 shows the differential between the left and right wheel speeds which is the selected method of control for turning the vehicle.

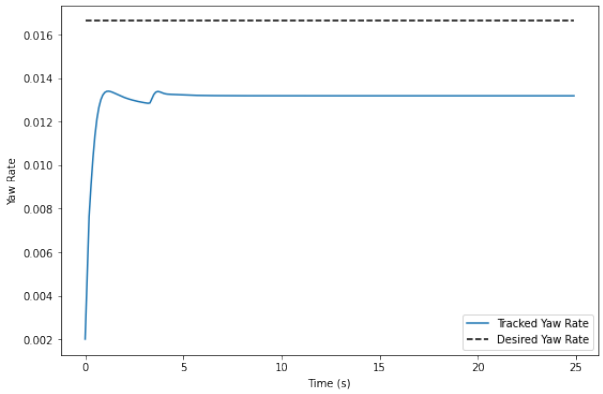

Fig. 2. Yaw Rate of the General EV

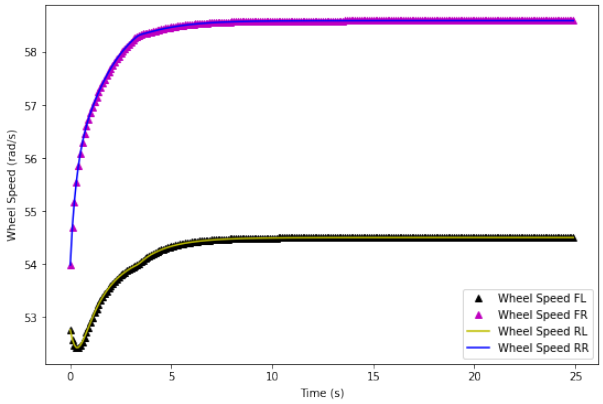

Fig. 3. Wheel Speed of the General EV

B. Dallara AV-21

The simulation run for the Dallara was a little different, resembling an overtaking maneuver. Due to the autonomous nature of this vehicle, the input is supplied in terms of checkpoints. Namely, at time 0, the vehicle is at the origin. The first checkpoint is at x = 150 m and y = -3 m. Then, the vehicle should slot back in front of the opponent vehicle at x = 350 m and y = 0. Moreover, to ensure robustness of our controller, we perform this test under perfect model knowledge, with a bank angle present and with a 5 offset in the states. The results of the three simulations are shown in figures 4,5 and 6.

In figure 4, we can observe that for all three cases, the maneuver is completed successfully. We further observe that model uncertainties slow down the MPC's performance, but the vehicle is still able to complete the overtake. Moreover, we see that with a 23 degree bank angle, the vehicle steers out much quicker, due to the additional stability provided by the presence of a bank angle.

In figure 5, we can clearly see that the vehicle slip angles are bounded to within 0.1 radians for all 4 wheels, which is our criteria for ensuring stability. Finally, figure 5 displays the load transfer between the left and right wheels during the overtake.

We observe that as the vehicle is turning out, the right tires carry more load and vice versa as it turns in. This is exactly what we expect for a typical vehicle under steering. Also, in the presence of a high bank angle, the left wheels always carry more load that the right wheels, which is perfectly expected. The oscillations in the normal loads could be decreased by tuning the MPC to put more weight on roll angle and roll rate.

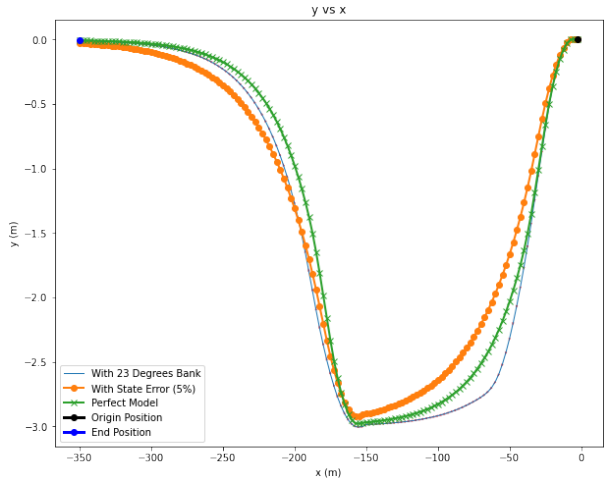

Fig. 4. Moving Trajectory of Dallara AV-21

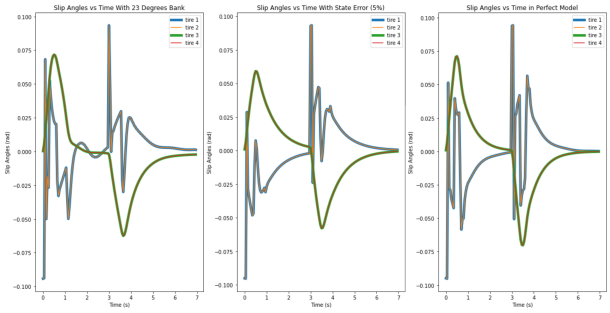

Fig. 5. Slip Angle of Dallara AV-21

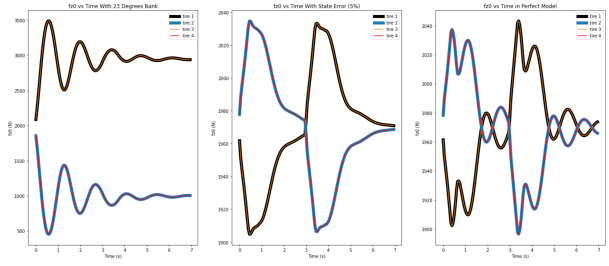

Fig. 6. fz0 of Dallara AV-21

VI. Conclusion

Ultimately, we can say that the MPC yielded satisfactory results for both a general electric vehicle and the Dallara AV-21. Through various simulations, we were able to test the effects of varying different parameters, as well as study the robustness of the controller, under the stability constraint points mentioned. We plan to continue working on this task through our Capstone project and the future work is outlined in the following sections.

VII. Future Work

A. Simulator

Our initial intended benchmark for this project was to use the deprecated ”cracked” open-source LGSVL simulator for the AV-21 MPC. This simulator has a medium-fidelity vehicle model for an AV-21 Dallara. A ROS node was completed and tied to a longitudinal MPC with optimized TTLS tuned to our MPCs coordinate frame. While the ROS node was stable, we ran into some compatibility issues with the LGSVL interfacing, and debugging this end-to-end was not completed in time. For future development, we will be completing and open-sourcing our CARLA simulator with a ChassisSim DLL back-end plugin for the AV-21 vehicle dynamics. With the intent to make the back-end vehicle dynamics modular and ”plug & play”. The time limitations were severe for this phase of our implementation.

B. Vehicle Model

Our future work will be to improve the model by incorporating a ChassisSim DLL vehicle model for the Dallara MPC. This DLL is a high-fidelity Dallara AV-21 model. Simulating the MPC in high-fidelity simulators and tuning the performance will provide better results than our current perfect model knowledge system.

C. Time Performance

Moreover, we have been working on developing a CasADi implementation of the MPC for the Dallara AV-21 using qpOASES to achieve maximum optimization for real-time implementation. While pyomo was easy to use it did not provide the necessary time optimization. Ultimately the MPC will be converted to C++ with a compatible solver for better performance.

References

[1] Ataei, M., Khajepour, A. and Jeon, S. (2019) “Model predictive control for integrated lateral stability, traction/braking control, and rollover prevention of electric vehicles,” Vehicle System Dynamics, 58(1), pp. 49–73. Available at: https://doi.org/10.1080/00423114.2019.1585557.

[2] Dizqah, A.M. et al. (2016) “A fast and parametric torque distribution strategy for four-wheel-drive energy-efficient electric vehicles,” IEEE Transactions on Industrial Electronics, 63(7), pp. 4367–4376. Available at: https://doi.org/10.1109/tie.2016.2540584.

[3] Dai, Y., Luo, Y. and Li, K. (2013) “Longitudinal and lateral coordinated motion control of four-wheel-independent drive electric vehicles,” 2013 World Electric Vehicle Symposium and Exhibition (EVS27) [Preprint]. Available at: https://doi.org/10.1109/evs.2013.6914736.

[4] Dai, Y., Luo, Y. and Li, K. (2013) “Longitudinal and lateral coordinated motion control of four-wheel-independent drive electric vehicles,” 2013 World Electric Vehicle Symposium and Exhibition (EVS27) [Preprint]. Available at: https://doi.org/10.1109/evs.2013.6914736.

Source Code:
GITHUB: ipynb Notebook Links to 4+ controller iterations
GITHUB: ROS Node created for AI Racing Tech AV-21
please contact ckwolfe@berkeley.edu if you experience Repo access issues.