Path-following and Adjustable Driving Behavior of Autonomous Vehicles using Dual-Objective Nonlinear MPC

Mohamed A. Daoud[1], Mostafa Osman[1], Mohamed W. Mehrez[1], William W. Melek[1]

***Abstract*— In this paper, a nonlinear model predictive control (NMPC) scheme is developed for the path-following control of autonomous vehicles. We introduce a dual-objective cost function which is composed of a regulation part and an economic part. By tuning the weights of the this cost, a driving behavior can be implemented; two different driving behaviors are designed, namely, energy-efficient and sport driving modes. A kinematic bicycle model is used for predicting the vehicle motion while a longitudinal motion dynamics model is used for estimating the energy consumption. Several real-time simulation scenarios are designed and implemented on Gazebo dynamic simulator. The results show the efficacy of the proposed controller.**

I. Introduction

Driving behavior of vehicles is a main concern in automotive industry and road traffic safety because of its effects on the environment in terms of energy consumption as well as the safety of other road users [1], [2]. Particularly, autonomous vehicles are expected to achieve better driving behavior compared to human driving to ensure both the safety of other road users, and the comfort of the vehicle passengers [3]. Moreover, this driving behavior should be adjusted according to the preference of the vehicle passengers.

Perception, localization, and control systems are among the the basis of autonomous vehicles research [4]. For control systems, Model Predictive Control (MPC) is widely used in autonomous vehicles since it can handle Multiple-Input Multiple-Output (MIMO) systems with state and input constraints [4]. MPC is a finite-horizon optimal-control method at which the control actions are calculated by iteratively solving an Optimal Control Problem (OCP) over a predefined horizon. The OCP is formulated given a cost function, which characterizes the control objective. Applied control actions are obtained by taking only the first element of an optimal control sequence, which results from solving the OCP [5]. For a review on the applications of MPC, see [6].

In [3], MPC for path tracking and collision avoidance for autonomous vehicles is used. The authors of [4] proposed using a kinematic model for prediction in MPC instead of using a dynamic model as the kinematic model showed a similar performance with the advantage of less computational effort. In [7], an MPC scheme for path-following that takes into consideration the road shape and the vehicle shape is proposed. Herein, a differential evolution algorithm is used to solve the optimization problem

[1]Mechanical and Mechatronics Engineering, University of Waterloo, ON, Canada {mohamed.daoud1, meaosman, mohamed.said, william.melek}@uwaterloo.ca

In [8], an MPC-based path-following controller is used to deal with system constraints. The goal of the proposed system is to improve path-following performance in case of uncertain tire-ground adhesion and road curvature conditions. A path-tracking controller based on MPC with a Laguerre function and exponential weight (LEMPC) is proposed in [9]. The optimization problem computation power is reduced by introducing a fitting orthogonal sequence consisting of Laguerre functions. The proposed method reduces the computational power while maintaining the tracking accuracy.

In [10], an NMPC scheme with a four-wheel vehicle prediction model is used to achieve obstacle avoidance on low-friction road surfaces. In [11], a collision-free path planner using MPC is introduced. The controller is only triggered when a collision is likely to happen. A scheme that uses NMPC to generate collision-free trajectory of obstacles is proposed in [12] , and used simplified vehicle dynamics to predict the vehicle state over the prediction horizon. In [13] a driving-behavior-aware stochastic method using MPC is proposed to minimize fuel consumption for hybrid electric buses.

The main contribution in this paper is the development of a control methodology that changes the driving behavior of autonomous vehicles using a dual-objective NMPC. The controller changes the driving behavior based on the passengers preference, while following a given reference path. This is accomplished by changing the weights of each term in the dual-objective cost function used in NMPC. A kinematic bicycle model is used as the prediction model of the NMPC controller. A real-time simulation model adapting the kinematic bicycle model is developed on Gazebo dynamic simulator [14]. Moreover, the proposed controller implementation is linked to the simulator using the Robot Operating System (ROS) [15]. The proposed controller performance is tested and validated with different road shapes.

The paper is organized as follows: the kinematic bicycle model and energy consumption model are presented in Section II. NMPC and the proposed dual-objective cost function are introduced in Section III. Then, the simulation environment and testing road scenarios are illustrated in Section IV. Results are presented and discussed in Section V. Finally, conclusions are drawn in Section VI.

Notations: in this paper, a diagonal matrix of the elements $x_1, x_2, ..., x_n$ is expressed as $diag(x_1, x_2, ..., x_n)$. $\mathbb{N}$ and $\mathbb{R}$ denote the natural and the real numbers, respectively. Moreover, $\mathbb{R}_{\geq 0}$ denotes the real numbers that are greater than or equal to zero. Finally, $||\cdot||_\tau^2$ refers to an l_2-norm weighted by τ.

978-1-7281-3473-4/19/$31.00 ©2019 IEEE

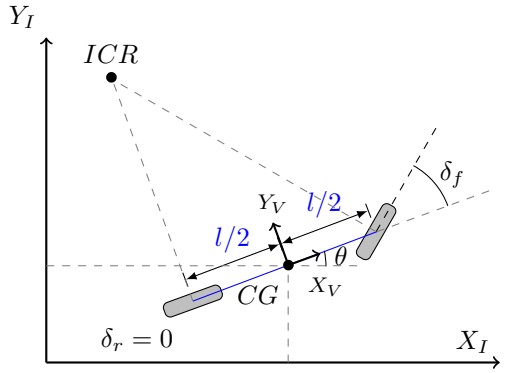


Fig. 1: Schematic diagram of the bicycle model, where frame I is the inertial frame and V is the vehicle local frame.

II. Bicycle and Energy Consumption Models

In this section, first we present the kinematic bicycle model used for the motion prediction of the vehicle. Next, we show the used vehicle energy consumption model.

A. Bicycle Model

The kinematic bicycle model used in this paper is derived by expressing wheels' rolling and sliding constraints. By combining the motion of each wheel under these constraints, the bicycle model is obtained [16]. This model expresses vehicle's motion in terms of the front wheel angular speed and its steering angle, which can be used to compute rear wheel speed. Fig. 1 shows a schematic diagram of the used bicycle model, where l is the vehicle's wheel base, θ is the heading angle relative to the inertial frame, CG is the center of gravity of the vehicle, ICR is the instantaneous center of rotation, and δ_f is the front wheel steering angle.

The discrete-time bicycle kinematic model can be written as follows:

$$\underbrace{\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}}_{\mathbf{x}_k} = \mathbf{f}_d(\mathbf{x}_{k-1}, \mathbf{u}_k)$$

$$= \mathbf{x}_{k-1} + t_s \underbrace{r\dot{\phi}_f \begin{bmatrix} \cos\delta_f \cos\theta - \frac{\sin\delta_f}{2} \sin\theta \\ \cos\delta_f \sin\theta + \frac{\sin\delta_f}{2} \cos\theta \\ \frac{\sin\delta_f}{l} \end{bmatrix}}_{\mathbf{f}_c(\mathbf{x}(t),\mathbf{u}(t))}, \quad (1)$$

where $\mathbf{f}_c$ is the continuous time model and $\mathbf{f}_d$ is its Euler discretization, k is the current time step, $\mathbf{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^\top \in \mathbb{R}^3$ is the state vector of the vehicle which is composed of the planar Cartesian coordinates of the center of the vehicle $[x, y]$ as well as the heading angle θ, and $t_s \in \mathbb{R}_{>0}$ is the sampling time. For the completeness of the mathematical formulation, the relation between rear wheel angular speed and front wheel angular speed is

$$\dot{\phi}_r = \dot{\phi}_f \cos(\delta_f).$$

B. Energy Consumption Modeling

In order to model the energy consumption in an electric ground vehicle, motor torque expressed in equation (2) is used.

$$T_m = \frac{r}{G}\left(ma + mgC_R + \frac{1}{2}\rho C_d A v^2\right) \quad (2)$$

where T_m is the electric motor torque, r is the wheel radius, G is the total reduction ratio, m is the vehicle mass, a is the vehicle acceleration, g is the gravitational acceleration, C_R is the rolling resistance, ρ is the air density, C_d is the drag coefficient, A is the frontal area of the vehicle, and v is the vehicle longitudinal speed. The motor torque equation (2) considers only motion in the longitudinal direction of the vehicle without taking into consideration the lateral forces. This is because the longitudinal motion consumes the major amount of energy in a ground vehicle.

Using equation (2), the total input energy $\mathcal{E}_{in}$ to the vehicle can be calculated through integrating the input power over time as shown below.

$$\mathcal{E}_{in} = \int_0^{t_f} \frac{T_m\ \omega}{\eta} dt, \quad (3)$$

where $\eta \in \mathbb{R}_{>0}$ represents the electric motor efficiency, ω is the angular speed of the electric motor in rad/sec, and t_f is the final time of the vehicles operation.

III. Proposed Dual-Objective NMPC

In this section, we start by explaining the path-following control problem and our proposed solution approach. Next, we formulate the optimal control problem (OCP) of the non-linear model predictive control (NMPC) scheme. Finally, we introduce the proposed dual-objective cost function utilized to change the behavior of the controller as discussed in Section I.

A. Path-following Control Problem

We are considering a path-following control using both vehicle's speed and steering as control inputs. Let us define the vehicle current state by $\mathbf{x}_k$, and the final target state by $\mathbf{x}_f$, where $\mathbf{x}_k, \mathbf{x}_f \in [\mathbf{x}_{min}, \mathbf{x}_{max}]$, and $\mathbf{x}_{min}$, $\mathbf{x}_{max}$ are the upper and lower limits of state $\mathbf{x}$, respectively. Moreover, we assume that $\mathbf{x}_k$ and $\mathbf{x}_f$ are points on a geometric path $\mathcal{S}$. Therefore, the path-following problem now requires that the vehicle state $\mathbf{x}_k$ moves along the path $\mathcal{S}$ towards the path end point $\mathbf{x}_f$, while the state and control constraints are satisfied.

Since the driving behavior changes based on the desired mode, $\mathcal{S}$ is not parameterized in time as the path timing itself is not known a priori [17]. Consequently, we parameterize

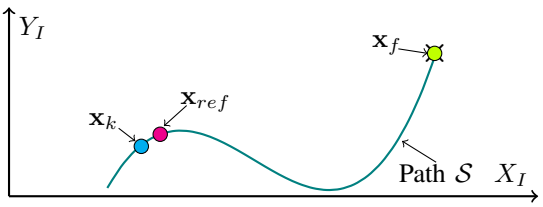


Fig. 2: A schematic of the path-following problem

$\mathcal{S}$ in a scalar path variable ε, i.e. $\mathcal{S} = s(\varepsilon)$. Now, the path-following problem is summarized in Alg. 1. In this algorithm, at each iteration, we calculate the actual path variable ε_{act} based on the current state $\mathbf{x}_k$ and the parametric representation of $\mathcal{S}$. Then, we add an increment $\psi \in \mathbb{R}$ to ε_{act} to obtain the reference path variable ε_{ref}. Finally, ε_{ref} is used to find the next reference state $\mathbf{x}_{ref}$. Here, we use κ as a path evolution step-size parameter that determines the smoothness of the transition along the path $\mathcal{S}$. Fig. 2 shows a schematic of the proposed path-following methodology.

Algorithm 1: Path-following

```
foreach time step k do
    Error = norm(x_f − x_k)
    if Error > Allowable_Error then
        ψ = κ − κ / (1 + Error)
        ε_act = s^{-1}(x_k)
        ε_ref = ε_act + ψ
        x_ref = s(ε_ref)
    else
        End of path reached
    end
end
```

Notice that the formulation of the increment ψ allows it to automatically decrease as the error between the current state and target state becomes smaller as shown below, i.e. $\lim_{\mathbf{x}_k \to \mathbf{x}_f} \psi = 0$.

B. NMPC Formulation

The constrained nonlinear model predictive control (NMPC) problem can be formulated as shown in equation (4). As can be seen in the equation, the problem is to minimize a cost function (introduced below) subject to the system kinematic model as well as the box constraints on the states, the controls, and the change in controls.

$$\min_{\mathbf{U}} \quad J(\mathbf{U}) \tag{4a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_{k-1}, \mathbf{u}_k), \quad \forall k \in [k, k+N] \tag{4b}$$

$$\Delta\mathbf{U}_{min} \leq \Delta\mathbf{U} \leq \Delta\mathbf{U}_{max}, \forall k \in [k, k+N-1] \tag{4c}$$

$$\mathbf{U}_{min} \leq \mathbf{U} \leq \mathbf{U}_{max}, \quad \forall k \in [k, k+N-1] \tag{4d}$$

$$\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}, \quad \forall k \in [k, k+N] \tag{4e}$$

In (4), $J : \mathbb{R}^{2\times N} \to \mathbb{R}_{\geq 0}$ is the cost function, $\mathbf{u}_k := [\dot{\phi}_f, \delta_f]^\top \in \mathbb{R}^2$ is the control action at time k, and $\mathbf{U} := \{\mathbf{u}_k, ..., \mathbf{u}_{k+N-1}\} \in \mathbb{R}^{2\times N}$ is the set of control actions over the prediction horizon length N, and $\mathbf{U}_{min}$, $\mathbf{U}_{max}$ are the upper and lower limits on $\mathbf{U}$, respectively. Also, $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$ is the change in the control action at time k, where for $k = 0$, $\Delta\mathbf{u}_0 = \mathbf{u}_0 - \mathbf{p}$ where $\mathbf{p}$ is the current measured wheel angular speed $\dot{\phi}_f$ and steering angle δ_f. $\Delta\mathbf{U} := \{\Delta\mathbf{u}_k, ..., \Delta\mathbf{u}_{k+N-1}\} \in \mathbb{R}^{2\times N}$ is the set of $\Delta\mathbf{u}_k$ over N, and $\Delta\mathbf{U}_{min}$, $\Delta\mathbf{U}_{max}$ are the upper and lower limits on $\Delta\mathbf{U}$, respectively.

C. Dual-Objective Cost Function

Here, we discuss our main contribution which is the use of a dual-objective cost function to consider different driving behaviors. By changing the weight of each element in the cost function, the NMPC solution differs in a way which incorporates a different controller behavior based on the passengers preference.

Furthermore, by adding an economic term that represents the energy consumption to the cost function, NMPC considers minimizing the power at each iteration. Therefore, the produced control actions reduce the needed driving power as well as the error to the reference signal.

To compute the total needed energy for a certain given path, equation (3) is used. To reduce the total consumed energy, either the motor efficiency or the output power, represented in torque and angular speed, can be added to the cost function. In this paper, we use the output power in the cost function. This showed an efficient and more comfortable driving behavior in our analysis. The proposed quadratic dual-objective cost function is given by:

$$J = \underbrace{\sum_{k=0}^{N} ||\mathbf{x}_{ref} - \mathbf{x}_k||_Q^2 + \sum_{k=0}^{N-1} ||\Delta\mathbf{u}_k||_R^2}_{J_r} + \underbrace{\sum_{k=0}^{N} ||T_m\ \omega||_E^2}_{J_e}, \tag{5}$$

where J_r is the regulating cost, J_e is the economic cost, $Q \in \mathbb{R}^{3\times3}$, $R \in \mathbb{R}^{2\times2}$, and $E \in \mathbb{R}$ are diagonal weighting matrices.

By reducing the weight of the output power in J_e, the NMPC controller focuses more on minimizing the error to the reference path signal and vice versa. Thus, we can obtain different driving behaviors by modifying the weights Q, R, and E.

IV. Dynamic Simulation Scenarios

The proposed controller is tested on a car model developed in Gazebo [14] simulator. The parameters of the model are for a Chevrolet Bolt car, see Table I. The NMPC controller is formulated using CasADi optimization library [18] and solved using the interior point method (IPOPT) [19]; the controller is implemented in Python programming language. The simulator and the controller are integrated under the Robot Operating System (ROS) [15]. The developed vehicle dynamic model satisfies Ackermann condition, i.e. when a vehicle is moving slowly on a curve, it turns slip-free [20]. Ackermann condition is given by:

$$\cot(\delta_o) - \cot(\delta_i) = \frac{w}{l}, \tag{6}$$

where δ_i and δ_o are the inner and the outer steering angle, respectively. The dimensions w and l are illustrated in Fig. 3.

TABLE I: Simulated Vehicle Parameters

Parameter	Value	Parameter	Value
m	1615 [kg]	G	7
l	2.6 [m]	A	2.0 [m^2]
w	1.5 [m]	ρ	1.2 [kg/m^3]
r	0.2 [m]	C_d	0.45
C_r	0.015	η	70% - 95%

Fig. 4 shows the developed ROS package. In this figure, $\dot{\phi}_{fi}$ and $\dot{\phi}_{fo}$ are the spinning speed of the front inner and

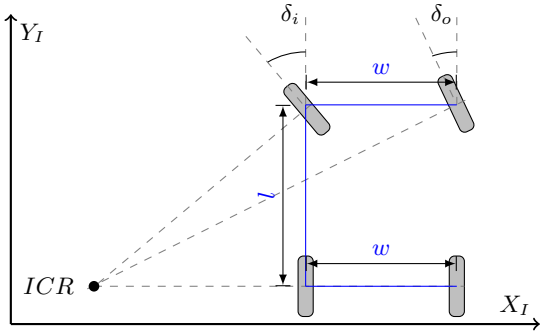


Fig. 3: Ackermann Model

outer wheels, respectively; and $\dot{\phi}_{ri}$ and $\dot{\phi}_{ro}$ are the spinning speed of the rear inner and outer wheels, respectively. We remark that this implementation is suitable for electric vehicles in which each wheel is driven independently (using in-wheel motors) without a main motor and a drive-line transmission.

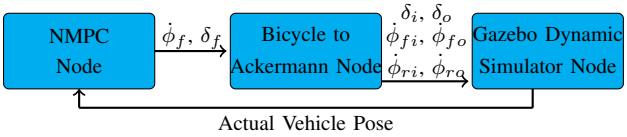
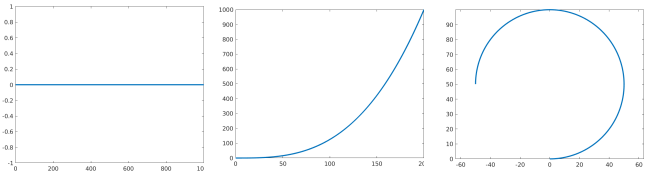


Fig. 4: The developed ROS package block diagram.

The reference paths used in our implementation are geometric splines simulating road and traffic lanes as suggested in [21]. We consider three main scenarios to test the proposed dual-objective NMPC in solving the path-following control problem given in Alg. 1. The reference paths in these scenarios are modeled using geometric equations as shown in Fig. 5.



(a) Straight Path
$\varepsilon \in [0, 1000]$
$x = \varepsilon$
$y = 0$

(b) Third Order Polynomial Path
$\varepsilon \in [0, 200]$
$x = \varepsilon$
$y = \frac{\varepsilon^3}{8000}$

(c) Circular Path
$\varepsilon \in [\frac{-\pi}{2}, \pi]$
$x = 50\cos(\varepsilon)$
$y = 50\ (1 + \sin(\varepsilon))$

Fig. 5: Different test cases

V. Real-time Simulation Results and Discussion

In this section, we present our real-time simulation results. We consider mainly two driving modes: energy-efficient driving mode, and sport driving mode. The two driving modes are tested with each reference path shown in Fig. 5. Here, we first present the tuning parameters of the proposed NMPC controller. Then, we present the simulation results for the considered driving scenarios.

A. Tuning parameters for NMPC

Tuning the cost function weights of each term is important in terms of NMPC performance. Table II shows the used parameters and weights, which led to the best path-following performance, for each test scenario. All simulations were conducted using a prediction horizon of $N = 20$ and a sample time of $t_s = 0.2$ [sec], $\mathbf{U}_{min} = [-15, \frac{-\pi}{6}]^\top$, $\mathbf{U}_{max} = [100, \frac{\pi}{6}]^\top$, $\Delta\mathbf{U}_{min} = [-8, -0.1]^\top$, and $\Delta\mathbf{U}_{max} = [4, 0.1]^\top$. The path evolution step-size parameter κ depends on the road shape. This is because it represents increments in meters in the first two cases, while it represents increments in radians in the third case, see Fig. 5.

TABLE II: Parameters and weights used for each test case

Driving Scenario	Params	Efficient Mode	Sport Mode
Straight Line	κ	30	30
	Q	$diag(4.75, 4.75, 100)$	$diag(4.75, 4.75, 100)$
	R	$diag(3, 20)$	$diag(1, 1)$
	E	2.5×10^{-10}	0
Spline Path	κ	20	20
	Q	$diag(10, 10, 2)$	$diag(10, 10, 2)$
	R	$diag(1, 1)$	$diag(1, 1)$
	E	2.5×10^{-10}	0
Circular Path	κ	0.157	0.157
	Q	$diag(100, 100, 0)$	$diag(100, 100, 0)$
	R	$diag(1, 1)$	$diag(1, 1)$
	E	2.5×10^{-10}	0

B. Straight Line Path

Most of city roads can be modeled as straight segments. In this test scenario, we consider the reference path in Fig. 5a; the path length is 1000 meter. Results obtained are showed in Fig. 6, 7 and 8 and summarized in Table III, where RMSE is the root mean squared error.

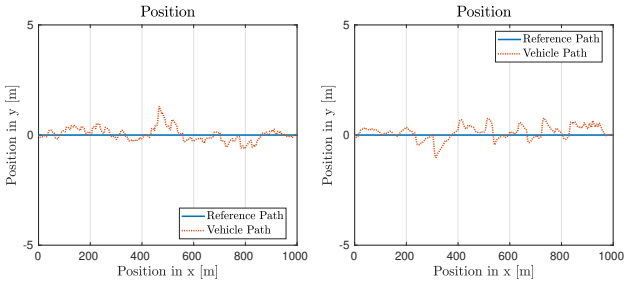


Fig. 6: Path simulation results for the straight line reference path. Left: Energy-efficient mode. Right: sport mode.

As can be seen in Fig. 7, the velocity profile shows slower acceleration and deceleration for the energy-efficient mode. Meanwhile, sport mode showed a higher speed velocity profile.

Fig. 8 shows the efficacy of the proposed controller in saving energy, while following the desired path, in case of the energy-efficient mode when compared to the sport mode.

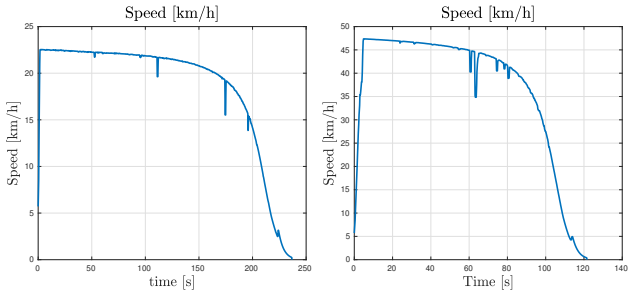


Fig. 7: The overall vehicle speed in km/h for both driving modes. Left: Energy-efficient mode. Right: sport mode.

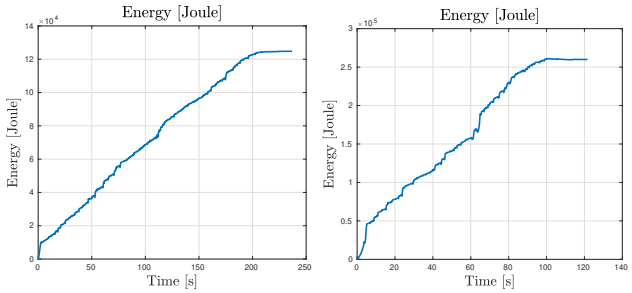


Fig. 8: Consumed energy (Joules) by the vehicle simulation results. Left: Energy-efficient mode. Right: sport mode.

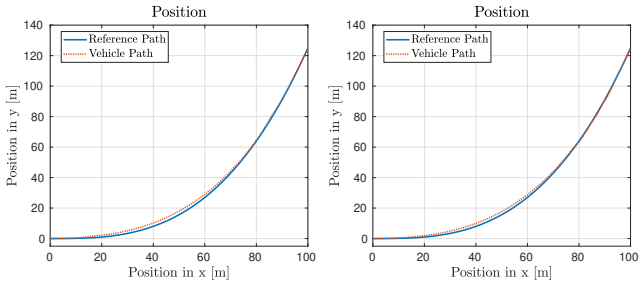


Fig. 9: Path simulation results for the spline reference path. Left: Energy-efficient mode. Right: sport mode.

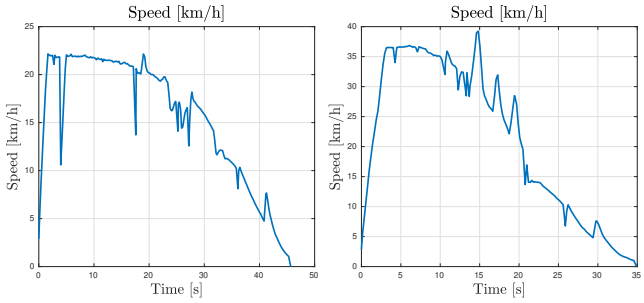


Fig. 10: The overall vehicle speed in km/h for both driving modes. Left: Energy-efficient mode. Right: sport mode.

TABLE III: Comparison between driving scenarios for straight line path

Parameter	Efficient Mode	Sport Mode
Total Energy Consumption	124.764 [kJ]	259.93 [kJ]
Total Travel Time	236.79 [sec]	121.62 [sec]
Average Speed	15.20 [km/h]	29.60 [km/h]
RMSE	0.2871 [m]	0.3161 [m]
End Point Error	0.0168 [m]	0.0017 [m]

The summary of the results in Table III shows that the energy-efficient mode was able to reduce the total consumed energy by 48% compared to the sport mode. Also, sport mode was able to arrive faster than the energy-efficient mode by 51.4%. The RMSE for both scenarios is very small. Small RMSE shows the effectiveness of the proposed controller in following the reference path.

C. Third Order Spline Path

Curved roads are very common cases especially in highways, and they can be modeled using polynomial splines. Now, we consider the reference path in Fig. 5b with a total length of 175.5 meter. Results obtained are showed in Fig. 9 and 10, and summarized in Table IV.

The results show a similar behavior compared to the straight line path. However, as this path is shorter, total consumed energy for both scenarios is less than that for the straight line path.

D. Circular Path

Finally, we use a circular reference path, Fig. 5c, to demonstrate the ability of the proposed controller in maintaining a fixed turning radius while driving. The total path length is 235.62 meter. Results obtained are showed in Fig. 11 and 12, and summarized in Table V.

TABLE IV: Comparison between driving modes for the spline path

Parameter	Efficient Mode	Sport Mode
Total Energy Consumption	26.86 [kJ]	44.2 [kJ]
Total Travel Time	45.67 [sec]	34.89 [sec]
Average Speed	13.83 [km/h]	18.11 [km/h]
RMSE	1.1335 [m]	0.9628 [m]
End Point Error	0.0512 [m]	0.0824 [m]

Similar to previous results, energy-efficient mode was able to reduce energy consumption, while sport mode was able to minimize total travel time.

The proposed controller maintained the input and state constraints throughout all simulations. Fig. 13 shows the computed steering angles for the spline path and the circular path for both driving modes. It can be noted from the figure that the control actions did not go beyond their limits. Path-following results for the three test cases are available at the following link: `https://youtu.be/2dSwTR1gk20`.

VI. Conclusion and Future Work

In this paper, we introduced a new path-following controller based on a dual-objective NMPC. The proposed controller adjusts the driving behavior of the autonomous vehicle based on the passenger preference. Two different controller behaviors are designed, namely, the energy-efficient and the sport modes. Moreover, a dual-objective cost function is used to shift between these two modes. In order to validate the proposed controller, different simulation scenarios are designed and implemented using Gazebo dynamic simulator. The results show the efficacy of the proposed controller.

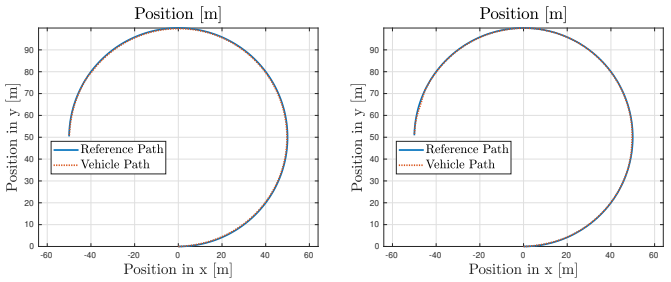


Fig. 11: Path simulation results for the circular reference path. Left: Energy-efficient mode. Right: sport mode.

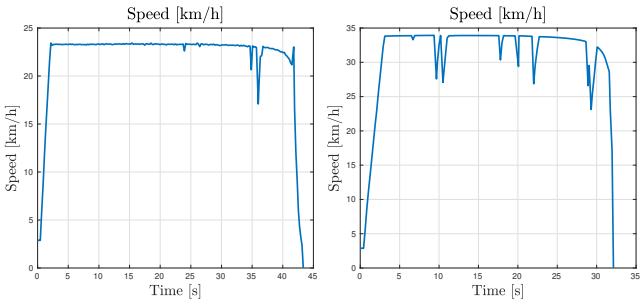


Fig. 12: The overall vehicle speed in km/h for both driving modes. Left: Energy-efficient mode. Right: sport mode.

The future work will focus on designing different driving behaviors beside the sport and energy-efficient modes. The stability of the proposed controller is to be rigorously proven with and without stabilizing constraints [5]. The effect of switching between different modes throughout vehicle operation will be studied in terms of the controller stability. Finally, the controller will be deployed on a real experimental platform for final experimental testing.

REFERENCES

[1] J. N. Barkenbus, "Eco-driving: An overlooked climate change initiative," *Energy Policy*, vol. 38, no. 2, pp. 762–769, 2010.

[2] T. Toledo and T. Lotan, "In-vehicle data recorder for evaluation of driving behavior and safety," *Transportation Research Record*, vol. 1953, no. 1, pp. 112–119, 2006.

[3] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2017.

[4] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1094–1099.

[5] L. Grüne and J. Pannek, "Nonlinear model predictive control," in *Nonlinear Model Predictive Control*. Springer, 2017, pp. 45–69.

[6] M. W. Mehrez, "Optimization Based Solutions for Control and State Estimation in Nonholonomic Mobile Robots: Stability, Distributed Control, and Localization," Ph.D. dissertation, Memorial University of Newfoundland, 2017.

[7] H. Guo, D. Cao, H. Chen, Z. Sun, and Y. Hu, "Model predictive path following control for autonomous cars considering a measurable disturbance: Implementation, testing, and verification," *Mechanical Systems and Signal Processing*, vol. 118, pp. 41–60, 2019.

[8] L. Yang, M. Yue, and T. Ma, "Path following predictive control for autonomous vehicles subject to uncertain tire-ground adhesion and varied road curvature," *International Journal of Control, Automation and Systems*, vol. 17, no. 1, pp. 193–202, 2019.

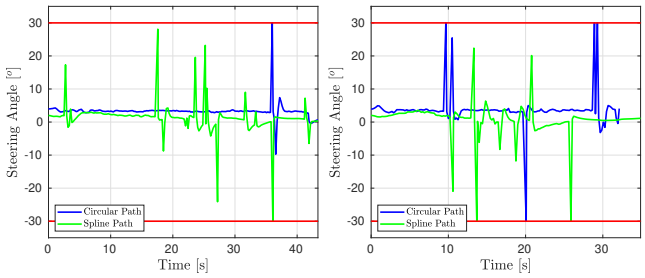


Fig. 13: Steering angle control action. Red lines represent the upper and lower limits. Left: Energy-efficient mode. Right: sport mode.

TABLE V: Comparison between driving scenarios for circular path

Parameter	Efficient Mode	Sport Mode
Total Energy Consumption	31.06 [kJ]	55.48 [kJ]
Total Travel Time	43.34 [sec]	32.13 [sec]
Average Speed	19.57 [km/h]	26.40 [km/h]
RMSE	0.4832 [m]	0.3395 [m]
End Point Error	0.0284 [m]	0.0015 [m]

[9] B. Zhang, C. Zong, G. Chen, and B. Zhang, "Electrical vehicle path tracking based model predictive control with a laguerre function and exponential weight," *IEEE ACCESS*, vol. 7, pp. 17 082–17 097, 2019.

[10] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear mpc algorithm for real-time obstacle avoidance of ground vehicles," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 4136–4141.

[11] T. Shim, G. Adireddy, and H. Yuan, "Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering*, vol. 226, no. 6, pp. 767–778, 2012.

[12] J. Park, D. Kim, Y. Yoon, H. Kim, and K. Yi, "Obstacle avoidance of autonomous vehicles based on model predictive control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 223, no. 12, pp. 1499–1516, 2009.

[13] L. Li, S. You, C. Yang, B. Yan, J. Song, and Z. Chen, "Driving-behavior-aware stochastic model predictive control for plug-in hybrid electric buses," *Applied Energy*, vol. 162, pp. 868–879, 2016.

[14] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[15] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[16] R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza, and R. C. Arkin, *Introduction to autonomous mobile robots*. MIT press, 2011.

[17] M. W. Mehrez, K. Worthmann, G. K. Mann, R. G. Gosine, and T. Faulwasser, "Predictive path following of mobile robots without terminal stabilizing constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9852–9857, 2017.

[18] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.

[19] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar 2006. [Online]. Available: https://doi.org/10.1007/s10107-004-0559-y

[20] R. N. Jazar, *Vehicle dynamics: theory and application*. Springer, 2017.

[21] A. Abramov, C. Bayer, C. Heller, and C. Loy, "A flexible modeling approach for robust multi-lane road estimation," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1386–1392.