LPV-MPC Path Planning for Autonomous Vehicles in Road Junction Scenarios

Luca Cavanini[1], Pawel Majecki[1], Mike J. Grimble[1], Vladimir Ivanovic[2], H. Eric Tseng[2]

Abstract - **The control of an autonomous host vehicle at a crossroads intersection, in the presence of uncoordinated target vehicles, and without any crossing priority regulation is considered. The problem was spilt into two sub-problems, namely the priority and the path-planning problems. These problems are solved using a hierarchical controller. The lower control level is a linear controller to control the vehicle's speed and heading, to follow the reference signals provided by a middle-level algorithm. This middle-level controller is a Model Predictive Control (MPC) computed by modelling a unicycle model that is in a Linear Parameter-Varying (LPV) state-space model form. Different features are introduced for improving the prediction capability of the LPV-MPC. Prediction data computed by the MPC are used by the higher-level state-machine supervisor algorithm to determine when the host vehicle can safely cross the junction. The hierarchical controller was tested in simulation using a set of stressing scenarios. Reported results show the effectiveness of the proposed LPV-MPC in managing complex traffic scenarios with efficient compute.**

I. INTRODUCTION

Recent engineering developments in automotive control systems for Autonomous Vehicles (AVs) suggest future transportation mobility will be significantly different and opportunities in energy consumption saving and traffic incidence reduction will arise, along with the availability of connectivity through vehicle to vehicle/infrastructure [1]. The AVs are considered the way forward to the future of mobility, because of the possibility of significantly reducing traffic, pollution and travel time [2]. In order to resolve the open challenges and achieve the required performance, different advanced control solutions for AVs have been proposed in the last few years [3] and each control policy has been designed for a particular scenario, e.g. highway travelling or coordinated traffic control [4].

In this paper the control of an autonomous Host Vehicle (HV) at a road junction in the presence of other uncoordinated Target Vehicles (TVs) is considered. In recent years, different solutions have been proposed to face this problem considering several aspects, such as crossing-time or the presence of traffic lights [5]. There has been a wide range of alternative control methods proposed in AV control systems for intersection handling, and two main themes are recurring: the introduction of a multi-level architecture of the control system [6] and the use of an optimal control approach [7] [8].

The use of a hierarchical control structure was formalized in [9] [10], presenting algorithms and features for different levels defined in the controller. The multi-level architecture is a common approach used in AV control problems [7] [8], but there is not a common paradigm for design using these different control levels. Because of the characteristics of the control problem, the use of predictive control algorithms appears a natural choice for developing an AV path-planner. Different predictive control solutions can be considered for facing the road intersection path-planning problem. These solutions involve different modelling methods (e.g. linearized models, time-varying models or nonlinear models) [11], optimization approaches [12] (e.g. quadratic, linear and nonlinear optimization) or predictive control paradigms [13] (e.g. robust or stochastic control).

In this paper, a hierarchical control system structure is designed in the following to exploit the features of the Model Predictive Control (MPC). The proposed multilevel control policy splits the road intersection control problem into sub-problems:

i) ***Priority problem:*** defining if the HV can safely cross the junction, according to the movement of other TVs and

ii) ***Path planning problem:*** defining the trajectory for crossing the junction, after the priority problem has been solved.

The priority problem is solved by a high level supervisor logic and the path planning problem is solved by middle and low level controllers. The *low level* involves a linear tracking controller, controlling the vehicle to follow the speed and heading reference signals provided by the middle-level. This *middle level* is an MPC-based path planner solving the planning problem by iteratively computing the HV optimal trajectory. The path planner uses a Linear Parameter-Varying (LPV)-MPC paradigm to reduce the computational time for the optimization problem, with reasonable tradeoff on diminished prediction accuracy [14]. Various features introduced to enhance the effectiveness of the LPV model approximation will be described in details and demonstrated in simulations. The path planner shares information with the supervisor for ensuring AV safety and increasing control performance. This is a key aspect, because in the considered scenario TVs ignore the AV behavior.

The paper is structured as follows. Section 2 presents the intersection handling problem. Section 3 describes the

[1]Luca Cavanini, Pawel Majecki and Mike J. Grimble are with Industrial Systems and Control Ltd., Glasgow, UK, {l.cavanini, pawel, m.grimble}@isc-ltd.com

[2]Vladimir Ivanovic and H. Eric Tseng are with Ford Research and Advanced Engineering, Dearborn, MI, USA, {vivanovi, htseng}@ford.com

proposed control system. Section 4 presents the simulation results and Section 5 summarizes the conclusions.

II. Problem Statement

In this section an unsignalized/uncontrolled intersection problem for HV in the presence of uncoordinated TVs is introduced as shown in Figure 1.

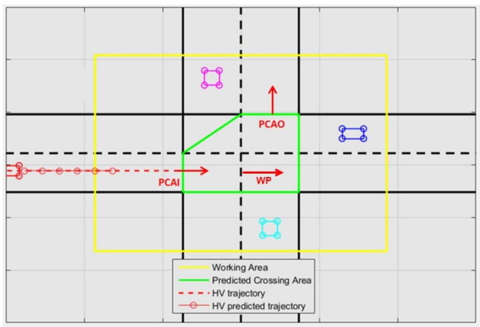

Figure 1. Road intersection scenario

The HV needs to cross the junction but avoiding the impact with other TVs. In the proposed control scenario, only three TVs have been considered. Because of their number and position with respect to the HV the results can be generalized to any possible road intersection scenario.

Remark 2.1. *TVs' future trajectories are estimated and updated every sampling time, and thus treated as known a priori information during the host controller's receding horizon.*

A. Vehicle Model

Modeling as a Dubins' car at the rear axle, the host vehicle's trajectory can be described by the following nonlinear system:

$$\begin{aligned}
\dot{x}_r &= v\cos(\alpha) \\
\dot{y}_r &= v\sin(\alpha) \\
x_f &= x_r + l_b\cos(\alpha) \\
y_f &= y_r + l_b\sin(\alpha) \\
\dot{\alpha} &= v\frac{\tan(\beta)}{l_b} \\
\dot{v} &= a
\end{aligned} \tag{1}$$

where (x_r, y_r) and (x_f, y_f) are rear and front wheel coordinates, v and a are longitudinal speed and acceleration, α and β are the orientation and steering angle and l_b is the distance between the rear and front wheels. . The control inputs are vehicle acceleration a and steering angle β. To ensure trajectory feasibility, we placed bounds on the controls where acceleration $\in (a_{min}, a_{max})$, steering angle $\in$ $(\beta_{min}, \beta_{max})$, steering rate $\in (\Delta\beta_{min}, \Delta\beta_{max})$ and limit the state where vehicle heading angular rate $\in (\Delta\alpha_{min}, \Delta\alpha_{max})$.

B. Road Intersection

The control scenario considered is shown in Figure 1. The three TVs considered are colored in blue, magenta and cyan. They are moving towards the junction to cross the intersection. The HV is the red vehicle on the left, moving to the junction. Several operating areas of the intersection are defined below to conduct the supervisor logic and control policy:

- Working Area (WA) is a convex hull that extends beyond the entire intersection, but limited to sensing capability (either by infrastructures or the HV).
- Predicted Crossing Area (PCA) is a convex polyhedron corridor inside the intersection connecting the initial position (PCAI) and desired destination (PCAO) of HV. It depends on the geometry of the intersection and the HV intention.
- PCA Initial set point (PCAI) is the pose (position and orientation) the vehicle must reach as it heads towards the intersection.
- PCA Over-the-junction set point (PCAO) is the pose (position and orientation) to be to be reached by the vehicle when the crossing operation is completed.
- Way Points (WPs) are intermediate poses defined between PCAI and PCAO in the PCA. A WP indicates a favourite pose of passage, for adapting the trajectory to a variety of road topologies (e.g. junctions of irregular shapes or roundabout).

Figure 1 shows WA and PCA in a left-turn road intersection scenario. The yellow square is the WA and the green shape is the PCA. This figure shows the instant where HV is approaching the intersection with its predicted trajectory shown in dashed line, it will reach PCAI and face the autonomous driving intersection handling control problem. The goal of the vehicle is to move through the PCA to reach PCAO, by passing through the WP.

III. Hierarchical Control System

In this section the hierarchal control system is designed for the intersection handling problem defined above. The architecture of the multi-level control system is shown in Figure 2.

As shown in Figure 2, the control system is organized in the three logical levels – Supervisor Logic, MPC Path Planner, and Speed & Attitude PID. The Supervisor Logic determines goal positions/poses; The MPC Path Planner plan and predict future poses of receding horizon with desired forward speed ($\bar{v}$) and heading angle ($\bar{\alpha}$); The low-level controller Speed & Attitude PID then track the desired velocity and heading by applying acceleration (a) and steering angle (β). The input command to the high-level Supervisor Logic is the Designated Lane $\in \{right, left, front\}$ to which the vehicle must move towards once it crossed the intersection. The Designated Lane is provided by an external controller that defines the vehicle's route over a given map [15]. At each sampling time, the Supervisor Logic provides Set-Points (a set of poses $x_{sp}(k+i), y_{sp}(k+i), \alpha_{sp}(k+i)$ for $i = 0,1,..,N$) to the mid-level controller where k indicates the current sampling time and the length of path planning horizon contains N samples.. The value of each pose in the Set-Points signal define spatial positions and orientations the vehicle will reach in the receding time horizon of length N. Note these Set-Points are the loosely populated goal positions/poses during the receding horizon, and not to be confused with the commonly used term 'waypoints' which describe intermediate points of vehicle positions/poses at a specific time on a the route of travel. Therefore, the superset of Set-Points only include the PCAI:=(x_I, y_I, α_I) (when the vehicle is approaching to the junction), the next WP:=(x_W, y_w, α_w) (if any, while the vehicle is moving through the junction,) and the PCAO:=(x_O, y_O, α_O) that is related to the Designated Lane at the end of the junction. Specifically, $\left(x_{sp}(k+i), y_{sp}(k+i), \alpha_{sp}(k+i)\right) \in \left[\left(x_j, y_j, \alpha_j\right)\right]$ where $j = \{I, W, O\}$ and $i = 0,1,..,N$. Each goal position Set-Point (x_j, y_j, α_j) is selected within the values $\{I, W, O\}$ by using the Prediction Data from the mid-level controller to determine feasible goals. Prediction Data contains predicted future poses of the HV (HV^P) computed by the mid-level controller $x_p(k+i), y_p(k+i), \alpha_p(k+i)$ with $i = 0,..,N$.

With the above outline, we will describe the details of all three level controllers, starting from the low level one.

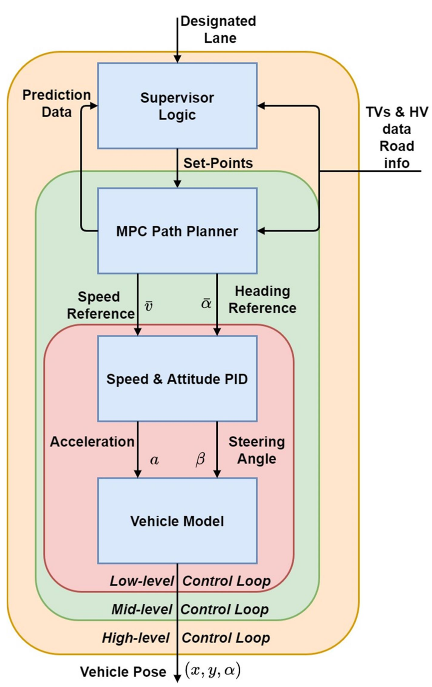

Figure 2. Hierarchical controller structure

A. *Low-Level Controller*

The low-level controller is designed to track the reference speed forward $\bar{v}$ and heading angle $\bar{\alpha}$ provided by the path planner by computing the forward acceleration a and the vehicle steering angle β. The lower-level controller is given by two Proportional-Integral (PI) controllers with a sample time T_L .

B. *Mid-Level Path Planner*

The path planner is formulated as a MPC to find the mid-level controls, i.e. desired vehicle speed $\bar{v}$, and heading $\bar{\alpha}$, that provide the optimal feasible path towards the Set-Points provided from Supervisor Logic. By assuming direct control of vehicle speed and heading, the Mid-Level Planner only needs to consider portion of Eq (1), that is,

$$\begin{aligned} v_x &= \bar{v}\cos(\alpha) \\ v_y &= \bar{v}\sin(\alpha) \\ \alpha &= \bar{\alpha} \end{aligned} \tag{2}$$

where v_x and v_y are the vehicle velocities expressed with respect to an external inertial reference frame and α is the vehicle heading. For defining the MPC control law, the continuous-time nonlinear model of the unicycle was transformed into a discrete-time LPV model, considering a sample time T_M. The discrete-time plant model used in the MPC design is as follows:

$$\begin{aligned} x(k+1) &= x(k) + \bar{v}(k)T_M\cos(\alpha(k)) \\ y(k+1) &= y(k) + \bar{v}(k)T_M\sin(\alpha(k)) \\ \alpha(k+1) &= \bar{\alpha}(k) \end{aligned} \tag{3}$$

As one may notice, this plant model is not linear and would typically pose a computational challenge when solving the optimal path. For computation efficiency, we will formulate it as a linear parameter varying system instead, where we assume the heading angle can be treated as a priori information and a scheduling parameter.

In practice, we will use the optimal path and heading solution over the receding horizon from the previously sampling time to schedule the LPV problem of current sampling time. Specifically, we will treat $cos\big(\alpha(k)\big)$ and $sin\big(\alpha(k)\big)$ as time-varying parameters, such that the plant is now a Linear Parameter Varying model and the scheduling parameters vector is $\rho(k) = \left[T_M\cos\big(\alpha(k)\big), T_M\sin(\alpha(k)k)\right]^T$. The LPV state-space matrices are evaluated at each time step iteratively given the value of the scheduling parameters. These are used in the MPC for predicting the outputs with $A_k = A(\rho(k))$, $B_k = B\big(\rho(k)\big)$ and $C_k = C(\rho(k))$, such that:

$$A_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}; B_k = \begin{bmatrix} \rho_1(k) & 0 \\ \rho_2(k) & 0 \\ 0 & 1 \end{bmatrix}; C_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

The LPV state-space model of the unicycle considers the state vector $\mathrm{x}_k = [x(k), y(k), \alpha(k)]^T$, the output vector $\mathrm{y}_k = \mathrm{x}_k$ and the control input vector $\mathrm{u}_k = [\bar{v}(k), \bar{\alpha}(k)]^T$. Starting from the Set-Points (goal/target positions) from the high level supervisor logic, the MPC Path Planner uses the previous LPV model for computing the predicted positions and orientations $(x_p(k+i), y_p(k+i), \alpha_p(k+i))$ by using the sequence of control signals obtained by solving the MPC problem presented in the following. These poses are computed by using the LPV design model and the previous time instance optimization sequence by following the same policy considered for computing predicted scheduling parameters and presented in the following. Predicted poses are collected into the Prediction Data signal to be passed to the Supervisor Logic for defining the Set-Points composing the MPC reference signal r. This is defined by the sequence of N Set-Points passed from the Supervisor Logic at the k-th time instance and stores predicted reference signals for the LPV-MPC, such that $\mathrm{r_k} = [\mathrm{r}_{k+1|k}, \dots, \mathrm{r}_{k+N|k}]^{\mathrm{T}}$ and $\mathrm{r}_{k+i/k} = [x_{sp}(k+i), y_{sp}(k+i), \alpha_{sp}(k+i)]^T$ with $i = 1, \dots, N_p$ ($\mathrm{r}_{k+i|k} = [x_j, y_j, \alpha_j]^T$ with $j \in \{I, W, O\}$).

Considering the previous LPV model and related signals, the MPC controller is formulated. When the cost-function is quadratic and constraints are affine, the LPV-MPC control law is obtained by solving the following optimization problem:

$$\begin{aligned}
&\min_{\mathrm{u}} \sum_{i=1}^{N_p} \| Q_y(\mathrm{y}_{k+i|k} - \mathrm{r}_{k+i|k}) \|_2^2 + \sum_{j=1}^{N_u} \| Q_u \mathrm{u}_{k+j|k} \|_2^2 \\
&s.t.\quad \mathrm{x}_{k+i+1|k} = A_{k+i|k}\mathrm{x}_{k+i|k} + B_{k+i|k}\mathrm{u}_{k+i|k} \\
&\mathrm{y}_{k+i} = C_{k+i|k}\mathrm{x}_{k+i|k} \\
&\mathrm{x}_{k|k} = \mathrm{x}_k \\
&\Delta\mathrm{u}_{\min} \le \Delta\mathrm{u}_{k+j|k} \le \Delta\mathrm{u}_{\max} \\
&\mathrm{u}_{\min} \le \mathrm{u}_{k+j|k} \le \mathrm{u}_{\max} \\
&\mathrm{y}_{\min} \le \mathrm{y}_{k+j|k} \le \mathrm{y}_{\max} \\
&N_u \le N_p \\
&\Delta\mathrm{u}_{k+i+1|k} = 0 \quad for \quad i \ge N_u
\end{aligned} \tag{5}$$

where $\boldsymbol{N_p}$ is the prediction horizon, $\boldsymbol{N_u}$ is the control horizon, $\boldsymbol{Q_y}$ and $\boldsymbol{Q_u}$ are square weighting matrices with $\boldsymbol{Q_u}$ invertible. The $\mathbf{x}_{\boldsymbol{k+i|k}}$ denotes the prediction of the variable $\mathbf{x}$ at time $\mathbf{k+i}$ based on the information available at time $\boldsymbol{k}$, $\boldsymbol{\Delta}\mathbf{u}_{\boldsymbol{k+i|k}}$ is the vector of input increments, with $\mathbf{u}_{\boldsymbol{k-1|k}} = \mathbf{u}_{\boldsymbol{k-1}}$, $\mathbf{r}_{\boldsymbol{k+i|k}}$ is the vector of output references, $(\boldsymbol{\Delta}\mathbf{u}_{\mathbf{min}}, \boldsymbol{\Delta}\mathbf{u}_{\mathbf{max}})$, $(\mathbf{u}_{\mathbf{min}}, \mathbf{u}_{\mathbf{max}})$, and $(\mathbf{y}_{\mathbf{min}}, \mathbf{y}_{\mathbf{max}})$ are values defining polyhedral sets of constraints on the input rate, and input and output signals, respectively.

The optimization sequence resulting from the solution of the MPC problem is then used for computing the predicted vehicle positions $x_p(k+i), y_p(k+i)$ and orientation $\alpha_p(k+i)$ with $i = [1, .. N_p]$, over the prediction horizon N_p. These are defined to be passed to the Supervisor by the Prediction Data signal and are computed by driving the LPV design model with the MPC optimization sequence, such that

$$\begin{aligned}
&\mathrm{x}_{k+1+i|k-i} = A_{k+i|k-1}\mathrm{x}_{k+i|k-1} + B_{k+i|k-1}\mathrm{u}_{k+i|k-1} \\
&\mathrm{u}_{k+i|k-i} = C_{k+i|k-1}\mathrm{x}_{k+i|k-1}
\end{aligned} \tag{6}$$

where $\mathrm{u}_{k+i|k-1}$ is the input for the predicted $k+i$-th prediction step computed in the previous time step k-1, the state-space matrices $A_{k+i|k-1}, B_{k+i|k-1}, C_{k+i|k-1}$ are the LPV model matrices evaluated at the $k+i$-th prediction time instance with respect to the previous time step k-1, and $\mathrm{y}_{k+i|k-1} = \mathrm{x}_{k+i|k-1} = [x_p(k+i), y_p(k+i), \alpha_p(k+i)]^T$ is the predicted pose of the vehicle at the future $k+i$-th prediction time instance with respect to the last control sequence computed at the previous k-1-th time instance. The state-space matrices can be considered constant, such that $A_{k+i|k-1} = A_{k|k-1}$, $B_{k+i|k-1} = B_{k|k-1}$, $C_{k+i|k-1} = C_{k|k-1}$ with $i = [1, .., N_p]$ if a scheduling parameter prediction policy is not considered.

Note: The above describes how the Prediction Data (predicted poses along the MPC prediction horizon) are used to schedule the system matrices parameters. The same Prediction Data $\left(x_p(k+i), y_p(k+i), \alpha_p(k+i)\right)$ is fed back to the High-Level supervisor logic for collision checking when traveling within a PCA. In the case where HV has yet to enter a PCA - for example, if the HV is slowing down towards PCAI - the high-level supervisor logic will use capable traveling speed v_{max} to assess safe-crossing instead of predicted speed. That is, set $\bar{v}(k+i) = v_{max}$ and $\bar{\alpha}(k+i) = 0$ with $i = 1, \dots, N_p$ as control sequence for computing predicted poses by Eq. (6). Such a set of predicted poses $\left(x_p(k+i), y_p(k+i), \alpha_p(k+i)\right)$ is included in Prediction Data signal if $\left(x_p(k+i), y_p(k+i), \alpha_p(k+i)\right) \in PCA$. The proposed LPV-MPC controller considers the standard form of the optimal control problem based the LPV model. To improve the effectiveness of the LPV model capability and the control performance, a number of different features have been considered:

1. Scheduling parameter prediction. With receding horizon control, where the control trajectory of the prediction horizon is updated/refined at each sampling time as the time progress, we will use the state trajectory predicted from the control trajectory of previous sampling time to approximate the system matrix. That is, the scheduling parameter of the LPV model is updated at each sampling time based on the predicted state (e.g. heading angle) along the prediction horizon in the previous sampling time. Specifically,

$$\begin{aligned}
A_{k+i|k} &\approx A_{k+i|k-1} = A(\rho(k+i\,|\,k-1)) \\
B_{k+i|k} &\approx B_{k+i|k-1} = B(\rho(k+i\,|\,k-1)) \\
C_{k+i|k} &\approx C_{k+i|k-1} = C(\rho(k+i\,|\,k-1))
\end{aligned} \tag{7}$$

where the predicted value of the scheduling parameter vector $\rho(k+i|k-1)$ is obtained from the predicted output $\mathrm{y}(k+i|k-1)$ with the optimal input sequence $\mathrm{u}_{k+i|k-1}$ computed in the previous time step, that is as in Eq.(6).

2. Convex approximation of the TVs constraints. The exact position of TVs is assumed to be known a-priori and can be used for defining the LPV-MPC problem. In practice, this assumed a-priori future TV positions (TV^P) are predicted based on constant speed and acceleration. While the prediction will not be accurate, the MPC planner will received updated TV information at each MPC Planner sampling time and adjust its constraints accordingly. To maintain the convexity of the optimization problem, an appropriate form of TV constraints is considered: in the case of only one TV, d_i is defined as the distance between the actual position of HV and the position of TV at the *i*-th prediction instance. HV and TVs have been represented by convex polygons and the distance between them is computed as the distance between two polygons, by exploiting convex polygons proprieties and according to the method proposed in [16]. Considering the set of distances d_i, with $i = 0,..,N_p$, between the HV actual position and future N_p positions of the TV, the minimum distance d_{min} is considered, such that:

$$d_{min} = \min\left(|d_1|, |d_2|, \dots, \left|d_{N_p}\right|\right) - \Delta s \tag{8}$$

This distance $d_{\min}$ is used to define a safe space around the HV, and it is constrained to move inside this space at the next time instant (and overall the prediction horizon):

$$\begin{aligned} \mathrm{x}_{\max}(k+i) &= \mathrm{x}(k) + d_{\min} \\ \mathrm{y}_{\max}(k+i) &= \mathrm{y}(k) + d_{\min} \\ \mathrm{x}_{\min}(k+i) &= \mathrm{x}(k) - d_{\min} \\ \mathrm{y}_{\min}(k+i) &= \mathrm{y}(k) - d_{\min} \end{aligned} \tag{9}$$

The policy neglects the presence of TV outside the Working Area (WA) or while HV is outside the WA. The distance d_{min} considers a security bound value (in the specific set to a fraction of the vehicle length, e.g. $\Delta s = 0.5l_v$) to improve the conservativeness and safety of constraints.

3. Soft and hard constraints on minimum forward speed. The proposed LPV-MPC considers a set of constraints on the minimum forward speed, such that the vehicle cannot assume a negative speed crossing the junction. These constraints cannot guarantee the vehicle does not stop while crossing the junction (because of the minimum velocity limit is $\bar{v}_{min0} = 0$). To reduce the possibility the HV may stop whilst crossing over the junction, two constraints on minimum speed ($\bar{v}_{minpos}$ and $\bar{v}_{min0}$) have been added, so that $\bar{v}_{minpos} > \bar{v}_{min} > 0$. A set of slack variables $s(k+i), \dots, s(k+N_u)$ have also been introduced for relaxing these constraints, such that:

$$\begin{aligned} \bar{v}_{minpos} &\le \bar{v}(k+i) + s(k+i) \\ \bar{v}_{min0} &\le \bar{v}(k+i) \end{aligned} \tag{10}$$

with $i = 1, \dots, N_u$. Slack variables are used in the cost-function with weights $Q_{\Delta u}(s) \gg Q_{\Delta u}(u)$, such that the optimization problem can set those variables $s(k+i) \neq 0$ only when the vehicle must be stopped to guarantee the feasibility of the problem. That is equivalent to stopping the vehicle to avoid a crash with road limits or TVs.

C. High-Level Supervisor Logic

The supervisor algorithm solves the priority problem (e.g. if host vehicle can enter the junction or PCA) by determining the set-point for the path planner. It is iteratively executed with a lower sampling rate ($T_H < T_M$) with respect to the path planner. If other TVs are crossing the junction as the host vehicle approaching the junction, Supervisor Logic would determine the goal position Set-Point as the position before entering the intersection (PCAI) – thus force HV to stop. Otherwise, the Set-Points are defined as the PCAO, or the next WP if any. Set-Points values are defined with respect to the Designated Lane, the position of the center of the junction (x_c, y_c) and the lane size L as in Table 1. While crossing the junction, the Set-Points signal vector components $(x_{sp}(k+i), y_{sp}(k+i), \alpha_{sp}(k+i))$ are iteratively defined by considering the achievable poses of the vehicle based on the Prediction Data obtained by Mid-Level Path Planner. The supervisor makes explicit use of the prediction computed by the MPC optimization sequence to determine goal positions Set-Points that are safe and efficient (or time competitive). The supervisor control logic is show in Figure 3.

D. Computational Complexity

Despite recent advance and technological innovation in the automotive control fields, the execution of advanced controllers on the most common Advanced Driver Assistance Systems (ADASs) would be a complex task. The proposed approach would permit to limit such issues by dividing the control target in two different problems that can be solved independently.

The priority problem is solved by the Supervisor Logic that permits to face any type of road intersection scenario according to a state-machine policy. This limits the computational complexity that can be evaluated a priori by considering the worst-case scenario the HV would face and overcoming computational issues related to other methods (e.g. solving priority and path planning problems in conjunction by single controller [17]).

The mid-layer of control has been formulated in the form of a LPV-MPC and would represent the main issues in terms of computational complexity. As presented in [14], the computational burden of MPC based on LPV model is strictly related to the time required for iteratively defining the optimization problem. On the other side, the optimization solver requires a not neglectable time to solve the MPC problem. The proposed LPV-MPC would be suitable to be formulated according to formulation permitting to limit such issues, e.g. by using a Multi-Parametric Programming [18] approach for reducing the optimization problem to an approximated and simpler form to be solved on-line, or by applying a nonlinear transformation to the original problem able to cast it in a simpler LTI-MPC [19].

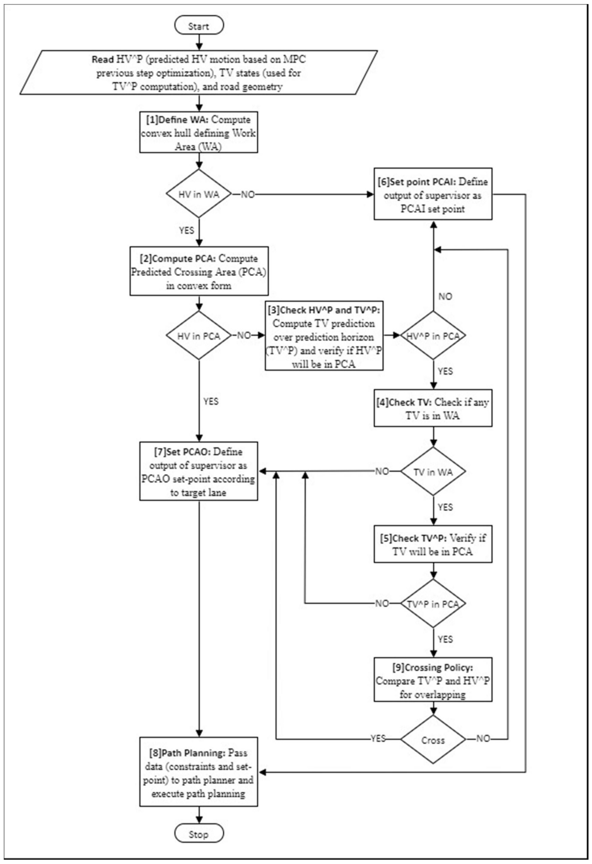

Figure 3. Supervisor Logic Diagram

Designated Lane	Set-Point
	$PCAI\!:= \left(x_c - L, y_c - \frac{L}{2}, 0\right)$
Right	$PCAO\!:= \left(x_c - \frac{L}{2}, y_c - L, -\frac{\pi}{2}\right)$
Front	$PCAO\!:= \left(x_c + L, y_c - \frac{L}{2}, 0\right)$
Left	$PCAO\!:= \left(x_c + \frac{L}{2}, y_c + L, \frac{\pi}{2}\right)$ $WP\!:= \left(x_c, y_c - \frac{L}{2}, 0\right)$

Table 1. Set-Points Definition

These solutions would permit to limit the computational complexity of the proposed MPC, further including the different features (e.g. scheduling parameters prediction) that would be integrated directly into any of the considered methods, with the possibility to extend the approach for including information provided from external systems/sensors (e.g. V2x communication).

IV. Simulation Results

In this section the simulation results obtained by testing the proposed control system for different junction traffic scenarios are reported. A right turn scenario (scenario 1) and a left turn scenario (scenario 2) have been considered. These scenarios have been characterized for testing the HV controller in stressing conditions. The MPC tuning parameters have been collected in Table 2.

The first scenario considers the HV facing a junction in the presence of three TVs approaching and moving over the junction, maintaining the initial lane during the test (each TV moves through the junction maintaining the initial heading). The control target of the HV is to move over the junction to reach the lane on the right over the junction. The HV a priori knows the instantaneous and future behavior of TVs, whereas TVs neglect the presence of the HV while performing the maneuver.

Symbol	Parameter	Value
N_u	Control Horizon	5
N_p	Prediction Horizon	6
N_{py}	Previewing Horizon	3
$N_{p\rho}$	Scheduling Parameters Prediction Horizon	3
Q_y	Output Weight Matrix (x, y, α)	diag([35 35 75])
Q_u	Input Weight Matrix $(\bar{v}, \bar{\alpha})$	diag([0.01 1])
u_{max}	Max Input Constraints $(\bar{v}, \bar{\alpha})$	(4.25, 4.71)
$u_{\min}$	Min Input Constraints $(\bar{v}, \bar{\alpha})$	(0, -4.71)
$\Delta u_{\min}$	Min Input Rate Constraints $(\Delta\bar{v}, \Delta\bar{\alpha})$	(-3.3527, -0.3142)
Δu_{max}	Max Input Rate Constraints $(\Delta\bar{v}, \Delta\bar{\alpha})$	(3.3527, 0.3142)
y_{min}, y_{max}	Min/Max Output Constraints	Given by road/TVs positions

Table 2. MPC Tuning Parameters

The initial conditions of the road traffic, while the HV is approaching the junction, are shown in Figure 4. The TVs initial positions are given in magenta, blue and cyan and the TV is represented by the red shape. Furthermore, the trajectory predicted over the prediction horizon at the initial time instant is represented by the dotted red line and the complete trajectory followed by the HV is given by the red dashed line. The WA is limited by a yellow line and the predicted crossing area used for defining the position constraints that the path planning algorithm should satisfy, is indicated by the green line. To evaluate the control performance and assess the effectiveness of the control solution, the dynamics of the input and output signals involved in control are as shown in Figure 5. The control effort computed by the MPC-based path planner is given by the reference heading and speed signals for controlling the low-level control loop. The vehicle speed dynamics involves linear behavior, so that the LPV-MPC computes the prediction over the horizon correctly and the vehicle output follows the velocity reference correctly. On the other hand, because of the nonlinear behavior of the HV heading dynamics, the reference angular position provided by the MPC does not exactly reflect the nonlinear system behavior and there is a delay between the reference and tracking results. The results relating to the heading angle are shown for the final heading set-point the vehicle should achieve over the junction (dash-dotted blue line) and the angular set-point considered by the MPC over the simulation (dotted blue line). This was defined according to the HV position by the supervisor. The MPC also considers constraints on the control input rate, acceleration and yaw rate dynamics. Input rate and magnitude constraints are represented by the magenta dash-dotted lines.

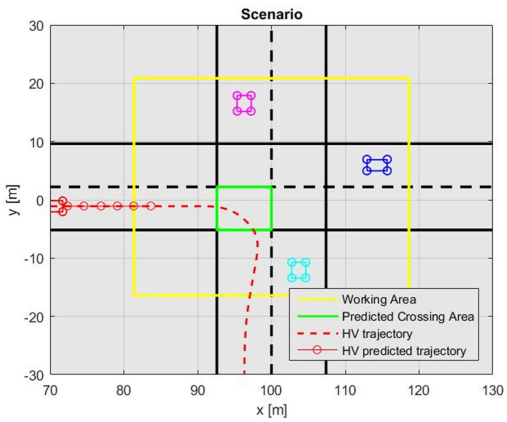

Figure 4. Traffic Scenario 1. Initial position of HV, TVs and HV trajectory.

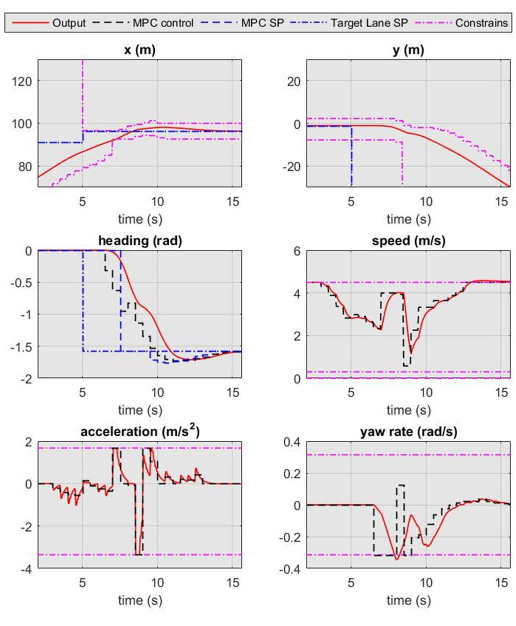

Figure 5. Traffic Scenario 1. Control Input and Controlled Output Signals

The HV performs the crossing operation while TVs are not in the PCA, and position constraints are given by the size and the shape of the PCA. Before crossing, the minimum position over the x-axis is computed with respect to the vehicle's position, whereas after the crossing the y-minimum position is constrained. While moving over a lane, later positions of the HV are constrained according to the lane size. The HV control speed is constrained by two different minimum values, representing the set of hard and soft constraints used for forcing the vehicle to avoid stopping during the junction crossing. The second scenario tested the algorithm while driving the HV for a left turn control target, assuming TVs act as in the previous test. The initial conditions giving the control scenario and the trajectory followed by the vehicle while moving through the junction are shown in Figure 6.

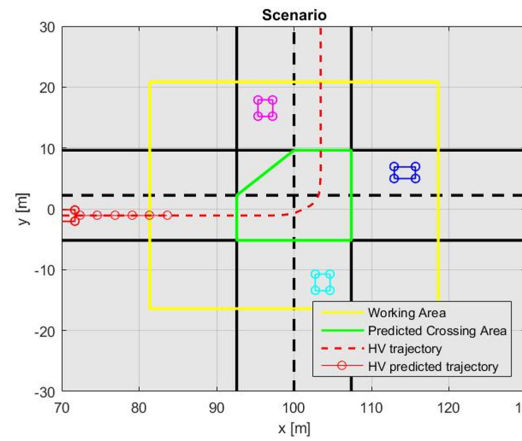

Figure 6. Traffic Scenario 2. Initial position of HV, TVs and HV trajectory.

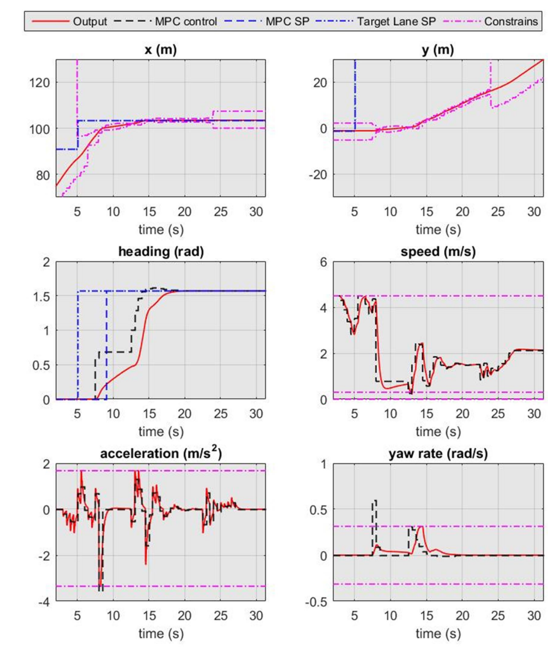

Figure 7. Traffic Scenario 2. Control Input and Controlled Output Signals

This shows the PCA convex set when approaching the left lane. Results showing the dynamics of the vehicle together with control input, controlled output and constraints are shown in Figure 7. While the HV is moving through the junction other TVs are in the PCA. These vehicles are considered for updating the position constraints considered by the MPC whilst computing the feasible trajectory.

Because of this, in the time interval for $7.5 < t < 22.5$ the value of max/min constraints for x and y linear position of the HV are close to the positions of the vehicle. When the vehicle is outside the junction, constraints are given by the lane size ant the position of the TVs moving in front on the HV, in the same line. Due to the presence of this vehicle, the velocity of the HV is limited while moving in the target lane ($t > 25$). Due to the anticipative action the heading trajectory defined by the MPC changes before the set-point signal. This is due to the use of preview on the heading reference signal ensures a feasible and smooth trajectory, avoiding overshoot during a change in direction. Further, the effect of the constraint softening given by activation of the slack variable is shown at $t = 7.5$ when the yaw rate computed by the MPC overcomes the constraints. Given the constraint softening feature, the problem maintains the feasibility, and the solution computed by the MPC can be computed over a relaxed set of solutions [6].

V. Conclusion

A Model Predictive Control (MPC) based path planning algorithm, working in conjunction with a supervisor logic and a commanded motion (speed/heading) following controller, has been proposed for solving the problem of an Autonomous Vehicle (AV) Host Vehicle (HV) crossing a road junction in the presence of uncoordinated Target Vehicles (TVs) which ignore the HV behavior.

The path planner was developed using an MPC framework and a Linear Parameter-Varying (LPV) representation of the nonlinear vehicle trajectory model. The LPV representation significantly reduced the computational burden for solving the original nonlinear optimal control problem with a certain degree of sub-optimality and approximation. The approximation error arose from the LPV modelling is further reduced by introducing various features detailed in this paper, leading to an effective solution of the original problem.

Supervisor logic has been developed to work in conjunction with the path planner to share information with it, so that the multi-modality of the intersection problem can be decoupled to logical decision making and convex path optimization. The safety margin of the vehicle whilst performing the crossing operation is thereby improved with judicious conservativeness (no two vehicles can occupy the same sub-areas of the intersection). Different traffic scenarios have been tested for the proposed solution.

The simulation results demonstrate the effectiveness of the controller in managing complex traffic scenarios, adapting the control action properly. Further research can be aimed at estimating the uncertainties of target vehicle behaviors, or developing an automated supervisor logic system, as opposed to handcrafted rules, for generalizing the applications to all possible traffic and road topology scenarios. Solutions based on Artificial Intelligence (AI) and Game Theory are potential candidates.

References

[1] Sun, Chao, Xinwei Shen, and Scott Moura., «Robust optimal eco-driving control with uncertain traffic signal timing,» in *2018 Annual American Control Conference (ACC)*, 2018.

[2] Axer, Steffen, and Bernhard Friedrich, «Estimating signal phase and timing for traffic actuated intersections based on low frequency floating car data,» in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016.

[3] Lim, Hansang, Wencong Su, and Chunting Chris Mi, «Distance-based ecological driving scheme using a two-stage hierarchy for long-term optimization and short-term adaptation,» in *IEEE Transactions on Vehicular Technology*, 2017.

[4] Sun, C., Shen, X., & Moura, S., « Robust optimal eco-driving control with uncertain traffic signal timing.,» in *Annual American Control Conference (ACC)*, 2018.

[5] Rosolia, Ugo, Stijn De Bruyne, and Andrew G. Alleyne, «Autonomous vehicle control: A nonconvex approach for obstacle avoidance,» *IEEE Transactions on Control Systems Technology,* pp. 469-484, 2016.

[6] Stone, P., Zhang, S., & Au, T. C., «Autonomous intersection management for semi-autonomous vehicles.,» in *Routledge Handbook of Transportation*, Routledge, 2015, pp. 116-132.

[7] Liu, Chang, Seungho Lee, Scott Varnhagen, and H. Eric Tseng, «Path planning for autonomous vehicles using model predictive control,» in *2017 IEEE Intelligent Vehicles Symposium (IV)*.

[8] Lee, Seungho, and H. Eric Tseng, «Trajectory Planning with Shadow Trolleys for an Autonomous Vehicle on Bending Roads and Switchbacks,» in *2018 IEEE Intelligent Vehicles Symposium (IV).*

[9] Tseng, H. Eric, «Hierarchical Automated Driving,» in *3rd IAVSD Workshop on Dynamics of Road Vehicles - Connected and Automated Vehicles*, Ann Arbor, Michigan, USA.

[10] Tseng, H. Eric, «Autonomous and Connected Vehicles,» in *Special Session, 2019 American Control Conference*, PHILADELPHIA, USA.

[11] Hult, R., Zanon, M., Gras, S., & Falcone, P. , «An MIQP-based heuristic for Optimal Coordination of Vehicles at Intersections,» in *IEEE Conference on Decision and Control (CDC)*, 2018 .

[12] Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E., «A survey of motion planning and control techniques for self-driving urban vehicles,» *IEEE Transactions on intelligent vehicles,* 2016.

[13] Du, Xinxin, Kyaw Ko Ko Htet, and Kok Kiong Tan, «Development of a genetic-algorithm-based nonlinear model predictive control scheme on velocity and steering of autonomous vehicles,» *IEEE Transactions on Industrial Electronics,* pp. 6970-6977, 2016.

[14] Cavanini, Luca, Gionata Cimini, and Gianluca Ippoliti, «Computationally efficient model predictive control for a class of linear parameter-varying systems,» *IET Control Theory & Applications,* pp. 1384-1392, 2018.

[15] Nolte, M., Rose, M., Stolte, T., & Maurer, M., «Model predictive control based trajectory generation for autonomous vehicles—An architectural approach,» in *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[16] K. Ball, «Convex geometry and functional analysis,» in *Handbook of the geometry of Banach spaces*, 2001, pp. 161-194.

[17] Huang, X., & Peng, H., « Speed trajectory planning at signalized intersections using sequential convex optimization,» in *American Control Conference (ACC)*, 2017.

[18] TøNdel, Petter, Tor Arne Johansen, and Alberto Bemporad, «An algorithm for multi-parametric quadratic programming and explicit MPC solutions,» *Automatica ,* vol. 39, n. 3, pp. 489-497, 2003.

[19] Cavanini, Luca, Gionata Cimini, and Gianluca Ippoliti, «A fast model predictive control algorithm for linear parameter varying systems with right invertible input matrix,» in *25th Mediterranean Conference on Control and Automation (MED)*, 2017.