

Application of Open-loop Stackelberg Equilibrium to Modelling a Driver's Interaction with Vehicle Active Steering Control in Obstacle Avoidance

Xiaoxiang Na and David J. Cole

Abstract—The increasing use of Active Front Steering (AFS) technology for obstacle avoidance raises the question of drivers' interaction with vehicle automation. Mathematical models capable of representing such interaction are in demand for driver behaviour study. This paper presents the application of open-loop Stackelberg equilibrium to modelling a driver's interaction with vehicle AFS control in an obstacle avoidance scenario, where both the driver and the AFS controller are exerting steering control to the vehicle. In this paper, such driver-AFS interactive steering control is modelled as a leader-follower game. Mathematical expressions of the driver's and the AFS controller's steering control strategies are derived using the Linear Quadratic Dynamic Optimization (LQDO) approach and the Distributed Model Predictive Control (DMPC) approach. These two approaches are found to give identical control gains, which suggests their equivalence in representing driver-AFS interaction. The DMPC approach is found to consume far less computation time due to its numerical nature. Mathematical modifications to the steering control strategies are then introduced to allow practical implementation for a future experimental study. Simulation results including time histories of steering angles and vehicle responses are illustrated and discussed.

Index Terms—Driver, Vehicle, Active Front Steering (AFS), Modelling, Game Theory

I. INTRODUCTION

THE use of Active Front Steering (AFS) technology [1] for assisting drivers in obstacle avoidance [2], [3] tended to increase the degree of complexity of the interaction between driver and vehicle. Since AFS technology allows a steering angle to be applied to vehicle road wheels independent of driver steering wheel angle, there is the possibility that the driver and the vehicle AFS controller compete for control, especially when the two hold different objectives. Such competition may jeopardize road safety. To this end, understanding drivers' interaction with vehicle active steering control becomes an important need. In vehicle engineering process, mathematical models able to represent driver steering control behaviour have been widely used to support vehicle product development [4]. However, little effort so far has been put in to model a driver's interaction with vehicle active steering control. It is expected

that such models may allow deeper insights into cognitive behaviours of human drivers so that optimization of present and future vehicle automation becomes a possibility.

Game theory has been widely used to model situations where two or more individuals make decisions that influence one another's welfare [5]. Of particular note here is the work done by Braun *et al.* [6] who constructed a noncooperative game model for predicting human players' hand movements in a rope-pulling game. Good agreement was found between model prediction and experimental measurements. This indicated that human sensorimotor interaction can be quantified in a game theoretic framework. Enlightened by this, game theory is used in this paper to represent a human driver's steering interaction with vehicle AFS control, where the driver and the AFS controller are modelled as game players. Particular focus is the application of Stackelberg equilibrium to the derivation of the driver's and the AFS controller's steering control strategies, with the information pattern restricted to be open-loop. In preparation for such work, definitions of strategy, Stackelberg equilibrium, leader-follower game, and information pattern are given in the next paragraph, followed by a discussion on the rationality of using Stackelberg equilibrium for investigating driver-AFS interactive steering control problem.

A strategy of a game player represents a mapping of the game's states to the player's action [7]. An equilibrium is a strategy set composed of each player's optimal strategy, i.e., the strategy maximizing the player's welfare [8]. A Stackelberg equilibrium is an equilibrium that emerges in a leader-follower game. In such a game, one player acts as the leader and the other players act as followers. The leader determines his or her strategy by taking into account all the followers' strategies whilst each of the followers responds to the leader's action by sticking to his or her optimal response [9]. Accordingly, a Stackelberg equilibrium is composed of the optimal strategies of the leader and the followers. These strategies are therefore named Stackelberg strategies. The information pattern of a game depicts players' knowledge of the states of the game [10]. The open-loop information pattern indicates that only the initial states of the game is known to players whilst closed-loop information pattern means that some of the intermediate states of the game are known. Such a difference can be illustrated using Fig. 1. This figure shows how a discrete system evolves as its players play games continuously. $\mathbf{w}(0)$ represents the

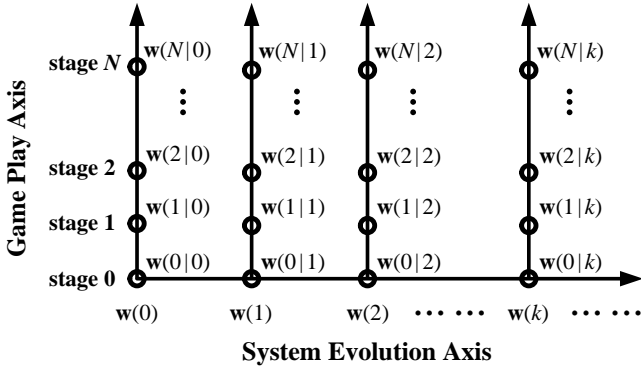


Fig. 1. Evolution of a discrete system with continuous game play

initial state of the system, i.e. the state at time step 0 along the system evolution axis. At this time, the players start to play a game. The game comprises a number of stages from stage 0 to N . The state of this game at stage 0 is expressed as $w(0|0)$ where the 0 to the right of the vertical bar symbol “|” represents time step whilst that to the left indicates game stage. At stage 0, each player applies an action to the system, causing the state of the game to transit to $w(1|0)$ and the game to move to stage 1. This then repeats over the following stages, causing the state of the game to transit in succession and finally to reach $w(N|0)$ at stage N . As a result of the N -stage game played, the state of the system evolves to $w(1)$ and time step 1 comes. At each future time step the players play a N -stage game, which pushes the system to evolve. At this point, a conclusive remark can be made that at an arbitrary time step k , the state of the system is $w(k)$, the initial state of the game at this time step is $w(0|k)$ which should equal $w(k)$, and the following intermediate states of the game can be expressed respectively as $w(j|k)$ for $j = 1, 2, 3, \dots, N$. With regard to the N -stage game played at time step k , the open-loop information pattern refers to the situation where players only know $w(0|k)$ whilst the closed-loop information pattern implies that the players have knowledge of some of the intermediate states $w(j|k)$.

The interaction between driver and vehicle AFS control in obstacle avoidance may be modelled as a leader-follower game. Assuming that an obstacle ahead of the vehicle was detected by an on-board sensing device, the vehicle AFS controller will plan a collision-free target path in response and, if necessary, apply steering angles to guide the vehicle to follow the target path, such as that described in [3] or [4]. However, the driver might try to control the vehicle to follow another path generated according to his or her cognition of the situation. If these two paths conflict, the driver and the AFS controller would possibly struggle for the governance of vehicle path following. Now, considering that the AFS controller employs a steering control strategy that partly neutralizes the driver’s steering wheel angle action, such as the one proposed by Anderson *et al.* [4], or similar to that developed by Tamaddoni *et al.* [11], the driver might react by taking the AFS controller’s strategy into account and compensating for its effects by applying additional steering wheel angle actions. In this case, the AFS controller bears close similarity to the follower in a leader-follower game whilst the

driver behaves in a comparable way to the leader.

Founded on the discussion made in the previous paragraph, the present paper addresses the application of Stackelberg equilibrium to the modelling of steering interaction between a driver and an AFS controller in a collision avoidance scenario, where the two controllers tend to follow different target paths. The driver and the AFS controller are respectively modelled as the leader and the follower of a leader-follower game, and their Stackelberg strategies for steering control are derived. The information pattern is restricted to be open-loop to reduce complications concerning intermediate game states. The present paper extends the authors’ initial study of the application of Stackelberg equilibrium to driver modelling [12] with a closer look at 1) the application of two mathematical approaches, i.e. Linear Quadratic Dynamic Optimization (LQDO) and Distributed Model Predictive Control (DMPC) to the derivation of the driver’s and the AFS controller’s theoretical Stackelberg steering control strategies, and 2) the necessary modifications to the theoretical strategies to allow experimental investigation of driver steering behaviour. To the authors’ knowledge, the research presented in this paper is amongst the first attempts to apply Stackelberg equilibrium in automotive engineering. Elsewhere in this area, there exist applications of Stackelberg equilibrium to energy management cases for hybrid electric vehicles, e.g. [13], [14].

The remainder of the article is organized as follows. Section II describes the driver-AFS interactive steering control scheme. Section III describes the use of LQDO and DMPC approaches to the derivation of the driver and AFS theoretical Stackelberg steering control strategies. Section IV presents comparisons of the numerical results from the two approaches. Section V explains modifications to the theoretical Stackelberg strategies for practical implementation. Section VI draws conclusions.

II. DRIVER-AFS INTERACTIVE STEERING CONTROL SCHEME

The driver-AFS interactive steering control scheme to be discussed throughout the paper is illustrated in Fig. 2. This scheme is identical to the one proposed in [12]. It outlines how a human driver might interact with an AFS controller via steering control in a leader-follower game.

Under such a scheme, the AFS controller is designed to determine its steering angle action $\delta_2(k)$ at time step k based on the AFS target path $\mathbf{R}_2(k)$, the vehicle state vector $\mathbf{x}(k)$, and the driver’s steering wheel angle $\delta_1(k)$. Conceptually, the AFS controller’s steering control strategy can be expressed as a function of $\mathbf{R}_2(k)$, $\mathbf{x}(k)$, and $\delta_1(k)$, such as

$$\delta_2(k) = f_2(\mathbf{R}_2(k), \mathbf{x}(k), \delta_1(k)) \quad (1)$$

$\mathbf{R}_2(k)$ can be planned by using various trajectory planning techniques, as described in [3]. $\mathbf{x}(k)$ comprises vehicle lateral velocity $v(k)$, yaw rate $\omega(k)$, lateral displacement $y(k)$ and yaw angle $\psi(k)$. $\mathbf{x}(k)$ can be sensed by motion sensors found in most commercialized vehicle dynamics control systems. $\delta_1(k)$ can be sensed by the AFS controller via an angle sensor. $f_2(\bullet)$ represents a rule that the AFS controller adopts for determining its steering angle $\delta_2(k)$. The $f_2(\bullet)$ discussed in

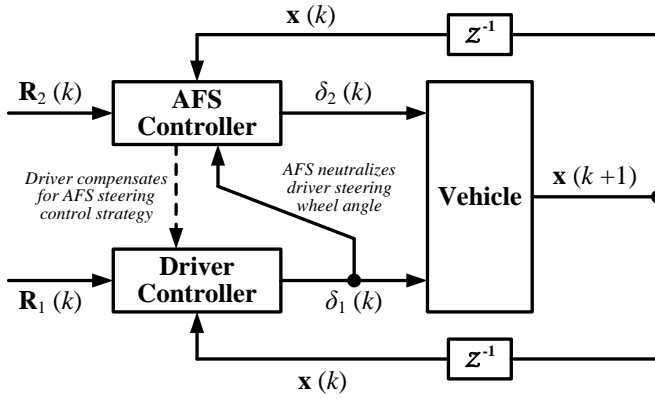


Fig. 2. Driver-AFS interactive steering control scheme

the present paper incorporates two aspects: 1) determination of a steering angle able to guide the vehicle to follow the AFS target path $\mathbf{R}_2(k)$, and 2) determination of an extra steering angle able to partially neutralize the driver's steering wheel angle $\delta_1(k)$. The latter aspect aims at mitigating possible negative impact of driver error in steering control. The AFS angle $\delta_2(k)$ determined according to (1) is eventually a combination of the angles resulted from the two aspects. Similar implementations have been reported by Anderson *et al.* [3] and Na and Cole [15], separately. In both cases, the AFS strategies were found effective in controlling a vehicle to avoid obstacle while neutralizing driver erroneous steering activities.

On the other hand, the driver is modelled to derive his or her steering control strategy through a cognitive process involving using his or her previewed road path $\mathbf{R}_1(k)$, sensed vehicle state vector $\mathbf{x}(k)$, and perceived AFS control strategy (1). Specifically, the driver is assumed to take into account the AFS steering control strategy (1) and intentionally compensates its effects via his or her own steering wheel control. Therefore, the driver's steering angle $\delta_1(k)$ satisfies the following equation:

$$\delta_1(k) = f_1(\mathbf{R}_1(k), \mathbf{x}(k), f_2(\mathbf{R}_2(k), \mathbf{x}(k), \delta_1(k))) \quad (2)$$

where $f_1(\bullet)$ represents a rule of mapping.

It can be seen from (1) that the AFS controller acts as the follower in the leader-follower game by simply responding to any driver steering angle $\delta_1(k)$. On the other hand, the driver acts as the leader who takes into account the AFS strategy (1) in deriving his or her own strategy. Therefore, the driver's and the AFS controller's optimal steering angle actions would satisfy:

$$\delta_1^*(k) = f_1^*(\mathbf{R}_1(k), \mathbf{x}(k), f_2^*(\mathbf{R}_2(k), \mathbf{x}(k), \delta_1^*(k))) \quad (3)$$

$$\delta_2^*(k) = f_2^*(\mathbf{R}_2(k), \mathbf{x}(k), \delta_1^*(k)) \quad (4)$$

where $\delta_1^*(k)$ and $\delta_2^*(k)$ respectively denote driver and AFS optimal steering angles, and $f_1^*(\bullet)$ and $f_2^*(\bullet)$ are corresponding mapping rules. Here the word "optimal" indicates that $\delta_1^*(k)$ and $\delta_2^*(k)$ respectively maximize the driver's and the AFS controller's individual welfare, or mathematically, minimize their individual cost functions. Details on the establishment of driver and AFS cost functions will be given in Section III.

By rearranging (3), $\delta_1^*(k)$ can be expressed conceptually as

a function of $\mathbf{x}(k)$, $\mathbf{R}_1(k)$ and $\mathbf{R}_2(k)$, that is

$$\delta_1^*(k) = s_1^*(\mathbf{x}(k), \mathbf{R}_1(k), \mathbf{R}_2(k)) \quad (5)$$

By substituting (5) into (4), $\delta_2^*(k)$ can be also expressed as a function of $\mathbf{x}(k)$, $\mathbf{R}_1(k)$ and $\mathbf{R}_2(k)$:

$$\delta_2^*(k) = s_2^*(\mathbf{x}(k), \mathbf{R}_1(k), \mathbf{R}_2(k)) \quad (6)$$

Following the definitions provided in Section I, (5) and (6) can be therefore named the Stackelberg steering control strategies, and $s_1^*(\bullet)$ and $s_2^*(\bullet)$ are the mapping rules. It should be noted that (5) and (6) are essentially theoretical Stackelberg strategies, where the word "theoretical" is used against its counterpart "practical". Detailed analytical expressions for (5) and (6) will be later provided in Section IV, where the differences between "practical" and "theoretical" strategies will also be explained.

It can be seen from Fig. 2 that the dynamics of the vehicle are under influence of both driver and AFS control. In this paper, the vehicle is modelled to operate in linear regime at constant speed to reduce the complexity of driver-AFS interaction. To this end, the linear time-invariant "bicycle" model described in [16] is used. The vehicle dynamics equation can be written as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (7)$$

where \mathbf{A} is the state matrix, \mathbf{B}_1 and \mathbf{B}_2 are respectively the input matrices associated with driver and AFS, and \mathbf{C} is the output matrix converting $\mathbf{x}(k)$ into output vector $\mathbf{z}(k)$. $\mathbf{z}(k)$ comprises vehicle lateral displacement $y(k)$ and yaw angle $\psi(k)$.

III. DERIVATION OF THEORETICAL STACKELBERG STEERING CONTROL STRATEGIES

In the Section II, both the driver's and the AFS controller's theoretical Stackelberg steering control strategy are written as functions of driver target path $\mathbf{R}_1(k)$, AFS target path $\mathbf{R}_2(k)$, and vehicle state $\mathbf{x}(k)$, as shown by (5) and (6). In this section, two approaches, namely the Linear Quadratic Dynamic Optimization (LQDO) and the Distributed Model Predictive Control (DMPC) are used to derive the detailed expressions of (5) and (6) under the open-loop information pattern. Numerical results produced by the two approaches are then compared.

A. Linear Quadratic Dynamic Optimization (LQDO)

The Linear Quadratic Dynamic Optimization (LQDO) approach is widely used to treat game theoretic problems where the environment system can be described by a linear differential equation and the objectives of players can be represented as functions containing just affine quadratic terms [17]. In this subsection, the procedure for taking the LQDO approach to the driver-AFS interactive steering control problem as illustrated in Fig. 2 is discussed. This involves three steps: construction of environment equation, establishment of cost functions, and derivation of theoretical Stackelberg strategies.

1) Environment Equation

In using the LQDO approach, an equation describing the environment system in which game players interact is required. Therefore this equation should be able to delineate both the evolution of vehicle dynamics and the updating of driver and AFS target paths. Following Sharp and Valtetsiotis [18], the driver's target path can be updated using a shift register:

$$\mathbf{R}_1(k+1) = \mathbf{A}_{r1}\mathbf{R}_1(k) + \mathbf{B}_{r1}\mathbf{r}_1^{\text{next}}(k) \quad (8)$$

where

$$\mathbf{R}_1(k) = \begin{Bmatrix} \mathbf{r}_1(0|k) \\ \mathbf{r}_1(1|k) \\ \vdots \\ \mathbf{r}_1(N_1^p|k) \end{Bmatrix}, \quad \mathbf{A}_{r1} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ \mathbf{0} & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_{r1} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \text{ and}$$

$$\mathbf{r}_1^{\text{next}}(k) = \mathbf{r}_1(N_1^p|k+1).$$

Here $\mathbf{R}_1(k)$ is the target path previewed by the driver at time step k . $\mathbf{R}_1(k)$ comprises a sequence of vectors from $\mathbf{r}_1(0|k)$ to $\mathbf{r}_1(N_1^p|k)$, denoting future target vehicle orientations up to the driver's preview horizon N_1^p . Each $\mathbf{r}_1(j|k)$ set ($j=0,1,2,\dots,N_1^p$) consists of two elements: target lateral displacement $r_1^y(j|k)$ and target yaw angle $r_1^\psi(j|k)$, that is

$$\mathbf{r}_1(j|k) = \begin{Bmatrix} r_1^y(j|k) \\ r_1^\psi(j|k) \end{Bmatrix}.$$

$\mathbf{r}_1^{\text{next}}(k)$ is a vector representing the target orientation set that will become $\mathbf{r}_1(N_1^p|k+1)$, i.e. the vehicle orientation set previewed N_1^p stages ahead at time step $k+1$. By repeating (8) the driver's target path $\mathbf{R}_1(k)$ can be updated in time.

Similarly, the AFS target path can be updated using:

$$\mathbf{R}_2(k+1) = \mathbf{A}_{r2}\mathbf{R}_2(k) + \mathbf{B}_{r2}\mathbf{r}_2^{\text{next}}(k) \quad (9)$$

$\mathbf{R}_2(k)$, \mathbf{A}_{r2} , \mathbf{B}_{r2} , $\mathbf{r}_2^{\text{next}}(k)$ and $\mathbf{r}_2(j|k)$ ($j=0,1,2,\dots,N_2^p$) take similar structures to their counterparts in (8). A note to make here is that the dimension of vector $\mathbf{R}_2(k)$ is N_2^p which denotes the AFS controller's preview horizon. N_2^p can be different from the driver's preview horizon N_1^p in theory.

Consequently, the environment system equation can be built up by merging target path updating equations (8) and (9) into vehicle dynamics equation (7), which gives:

$$\mathbf{w}(k+1) = \mathbf{A}_w\mathbf{w}(k) + \mathbf{B}_{w1}\delta_1(k) + \mathbf{B}_{w2}\delta_2(k) + \mathbf{F}\mathbf{R}^{\text{next}}(k) \quad (10)$$

where

$$\mathbf{w}(k) = \begin{Bmatrix} \mathbf{x}(k) \\ \mathbf{R}_1(k) \\ \mathbf{R}_2(k) \end{Bmatrix}, \quad \mathbf{R}^{\text{next}}(k) = \begin{Bmatrix} \mathbf{r}_1^{\text{next}}(k) \\ \mathbf{r}_2^{\text{next}}(k) \end{Bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix},$$

$$\mathbf{A}_w = \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{r1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{r2} \end{bmatrix}, \quad \mathbf{B}_{w1} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \text{ and } \mathbf{B}_{w2} = \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

2) Cost Functions

In the driver-AFS interactive steering control problem, the driver at each time step k attempts to control the vehicle to track his or her target path $\mathbf{R}_1(k)$ while considering the amount of steering effort. Hence, the driver's objective can be formulated as to minimize a linear quadratic cost function that penalizes both vehicle path-following errors along $\mathbf{R}_1(k)$ and his or her steering wheel angle actions. By adopting the style used by Lewis *et al.* [19], the driver's cost function can be written as:

$$J_1(k) = q_1^y \Delta_1^y(N_1^p|k)^2 + q_1^\psi \Delta_1^\psi(N_1^p|k)^2 + \sum_{j=0}^{N_1^p-1} [q_1^y \Delta_1^y(j|k)^2 + q_1^\psi \Delta_1^\psi(j|k)^2 + \delta_1(j|k)^2] \quad (11)$$

where $\Delta_1^y(j|k)$ and $\Delta_1^\psi(j|k)$ ($j=0,1,2,\dots,N_1^p$) respectively denote the lateral displacement and yaw angle errors evaluated by the driver along $\mathbf{R}_1(k)$ at time step k , that is,

$$\Delta_1^y(j|k) = y(j|k) - r_1^y(j|k) \quad (12a)$$

$$\Delta_1^\psi(j|k) = \psi(j|k) - r_1^\psi(j|k) \quad (12b)$$

and q_1^y and q_1^ψ are corresponding path-error weights.

The AFS controller can be designed to hold a similar cost function that weighs its path-following errors and steering control actions, such as the one proposed in [3] or [20]. Hence, the AFS controller's cost function can be expressed as:

$$J_2(k) = q_2^y \Delta_2^y(N_2^p|k)^2 + q_2^\psi \Delta_2^\psi(N_2^p|k)^2 + \sum_{j=0}^{N_2^p-1} [q_2^y \Delta_2^y(j|k)^2 + q_2^\psi \Delta_2^\psi(j|k)^2 + \delta_2(j|k)^2] \quad (13)$$

where

$$\Delta_2^y(j|k) = y(j|k) - r_2^y(j|k) \quad (14a)$$

$$\Delta_2^\psi(j|k) = \psi(j|k) - r_2^\psi(j|k) \quad (14b)$$

and q_2^y and q_2^ψ are corresponding path-error weights.

It is worth noting that the essentials of a leader-follower game were not embodied in either of the two steps described above, i.e. construction of environment equation and establishment of cost functions. In other words, these two steps are in general use for several categories of driver-AFS interaction issues, as described in [12]. It is the methods by which cost functions (11) and (13) are optimized to derive the theoretical Stackelberg strategies that will distinguish the essentials of a leader-follower game. This will be explained in the following few paragraphs.

3) Theoretical Stackelberg Strategies

Since the information pattern is restricted to be open-loop in the present paper, the derivation of open-loop Stackelberg strategies reported by Hungerländer and Neck [21] is followed, adapting to the driver-AFS interactive steering control problem discussed. In preparation for the derivation, the driver's and the AFS controller's cost functions (11) and (13) are rewritten to (15) and (16) respectively:

$$J_1^{\text{LQDO}}(k) = \frac{1}{2} \mathbf{w}^T(N_1^p | k) \mathbf{H}_1^T \mathbf{Q}_1 \mathbf{H}_1 \mathbf{w}(N_1^p | k) + \frac{1}{2} \sum_{j=0}^{N_1^p-1} [\mathbf{w}^T(j | k) \mathbf{H}_1^T \mathbf{Q}_1 \mathbf{H}_1 \mathbf{w}(j | k) + \delta_1(j | k)^2] \quad (15)$$

$$J_2^{\text{LQDO}}(k) = \frac{1}{2} \mathbf{w}^T(N_2^p | k) \mathbf{H}_2^T \mathbf{Q}_2 \mathbf{H}_2 \mathbf{w}(N_2^p | k) + \frac{1}{2} \sum_{j=0}^{N_2^p-1} [\mathbf{w}^T(j | k) \mathbf{H}_2^T \mathbf{Q}_2 \mathbf{H}_2 \mathbf{w}(j | k) + \delta_2(j | k)^2] \quad (16)$$

where

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & -1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix},$$

$$\mathbf{H}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & \cdots & 0 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 & -1 & 0 & \cdots & 0 \end{bmatrix},$$

$$\mathbf{Q}_1 = \begin{bmatrix} q_1^y & 0 \\ 0 & q_1^{\psi} \end{bmatrix}, \text{ and } \mathbf{Q}_2 = \begin{bmatrix} q_2^y & 0 \\ 0 & q_2^{\psi} \end{bmatrix}.$$

Here \mathbf{H}_1 is a matrix that transforms each intermediate environment state $\mathbf{w}(j | k)$ ($j=0, 1, 2, \dots, N_1^p$) into $\mathbf{H}_1 \mathbf{w}(j | k)$ which consists of vehicle lateral displacement error $\Delta_1^y(k)$ and yaw angle error $\Delta_1^{\psi}(k)$ from the driver's perspective, that is

$$\mathbf{H}_1 \mathbf{w}(j | k) = \begin{bmatrix} \Delta_1^y(j | k) \\ \Delta_1^{\psi}(j | k) \end{bmatrix} = \begin{bmatrix} y(j | k) - r_1^y(j | k) \\ \psi(j | k) - r_1^{\psi}(j | k) \end{bmatrix}.$$

\mathbf{Q}_1 is the weight matrix for the driver's path-following errors. \mathbf{H}_2 and \mathbf{Q}_2 are corresponding matrices for the AFS controller.

Based on the analysis provided by Hungerländer and Neck [21], the optimization problem of the AFS controller (follower) at time step k can be formulated as:

$$\min_{\delta_2} J_2^{\text{LQDO}}(k) \quad \text{s.t. } \mathbf{w}(j+1 | k) = \mathbf{A}_w \mathbf{w}(j | k) + \mathbf{B}_{w1} \delta_1(j | k) + \mathbf{B}_{w2} \delta_2(j | k) + \mathbf{F}^{\text{next}}(j | k) \text{ for } j = 0, 1, 2, \dots, N_2^p - 1 \quad (17)$$

where the constraint is the equation governing the progression of the game played at time step k , or in other words, the game progression equation at time step k . It can be seen that this equation takes the same form as the environment system equation (10). The optimization problem (17) can be treated by using the Lagrange multiplier method [19] which starts with constructing the augmented cost function (18) with respect to the AFS controller's original cost function (16):

$$J_2^{\text{LQDO}}(k) = \frac{1}{2} \mathbf{w}^T(N_2^p | k) \mathbf{H}_2^T \mathbf{Q}_2 \mathbf{H}_2 \mathbf{w}(N_2^p | k) + \frac{1}{2} \sum_{j=0}^{N_2^p-1} [\mathbf{w}^T(j | k) \mathbf{H}_2^T \mathbf{Q}_2 \mathbf{H}_2 \mathbf{w}(j | k) + \delta_2(j | k)^2] + \frac{1}{2} \sum_{j=0}^{N_2^p-1} \{ \mathbf{p}^T(j+1 | k) [\mathbf{A}_w \mathbf{w}(j | k) + \mathbf{B}_{w1} \delta_1(j | k) + \mathbf{B}_{w2} \delta_2(j | k) + \mathbf{F}^{\text{next}}(j | k) - \mathbf{w}(j+1 | k)] \} \quad (18)$$

In (18), each $\mathbf{p}(j+1 | k)$ where $j=0, 1, 2, \dots, N_2^p - 1$ denotes a Lagrange multiplier at a particular stage of the game played at time step k . It can be seen that each $\mathbf{p}(j+1 | k)$ is associated with an expression derived from the game progression equation. Hence the augmented cost function (18) equals its origin (16). (18) reaches its minimum when its partial derivatives with respect to all its variables equal zero, that is

$$\frac{\partial J_2^{\text{LQDO}}(k)}{\partial \mathbf{w}(j | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N_2^p - 1, \text{ and } \frac{\partial J_2^{\text{LQDO}}(k)}{\partial \mathbf{w}(N_2^p | k)} = \mathbf{0} \quad (19a)$$

$$\frac{\partial J_2^{\text{LQDO}}(k)}{\partial \delta_2(j | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N_2^p - 1 \quad (19b)$$

$$\frac{\partial J_2^{\text{LQDO}}(k)}{\partial \mathbf{p}(j+1 | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N_2^p - 1 \quad (19c)$$

As a result, the necessary conditions for a minimum (18), or equivalently, a minimum (16) are:

$$\mathbf{p}(j | k) = \mathbf{H}_2^T \mathbf{Q}_2 \mathbf{H}_2 \mathbf{w}(j | k) + \mathbf{A}_w^T \mathbf{p}(j+1 | k)$$

for $j = 0, 1, 2, \dots, N_2^p - 1$, with terminal condition

$$\mathbf{p}(N_2^p | k) = \mathbf{H}_2^T \mathbf{Q}_2 \mathbf{H}_2 \mathbf{w}(N_2^p | k) \quad (20a)$$

$$0 = \delta_2(j | k) + \mathbf{B}_{w2}^T \mathbf{p}(j+1 | k) \text{ for } j = 0, 1, 2, \dots, N_2^p - 1 \quad (20b)$$

Three notes shall be made here. Firstly, the calculation involved in (19a) needs determination of $\partial \delta_1(j | k) / \partial \mathbf{w}(j | k)$ and $\partial \delta_2(j | k) / \partial \mathbf{w}(j | k)$ over $j = 0, 1, 2, \dots, N_2^p - 1$. Since the open-loop information pattern was assumed, only $\mathbf{w}(0 | k)$, or equivalently $\mathbf{w}(k)$ is known to the driver and the AFS controller whilst none of the intermediate states $\mathbf{w}(j | k)$ for $j = 1, 2, 3, \dots, N_2^p - 1$ will be known, as explained in Section I. Hence, both $\partial \delta_1(j | k) / \partial \mathbf{w}(j | k)$ and $\partial \delta_2(j | k) / \partial \mathbf{w}(j | k)$ become zero over $j = 1, 2, 3, \dots, N_2^p - 1$. As a result, calculation of the partial derivatives involved in (19a) leads to (20a). Secondly, the calculation involved in (19b) gives (20b). Finally, the calculation involved in (19c) yields the game progression equation (17) and it is therefore not repeated here.

Continuing to follow Hungerländer and Neck [21], the optimization problem of the driver (leader) can be expressed as:

$$\min_{\delta_1} J_1^{\text{LQDO}}(k) \quad \text{s.t. } \mathbf{w}(j+1 | k) = \mathbf{A}_w \mathbf{w}(j | k) + \mathbf{B}_{w1} \delta_1(j | k) + \mathbf{B}_{w2} \delta_2(j | k) + \mathbf{F}^{\text{next}}(j | k) \text{ for } j = 0, 1, 2, \dots, N_1^p - 1, \text{ and equation (20a), and equation (20b)} \quad (21)$$

where it can be seen that (20a) and (20b), that is, the necessary conditions for minimizing the AFS controller's cost function, serve as the constraints. This correlates to the presumption that in a leader-follower game the driver (leader) takes into account the AFS steering control strategy in deriving his or her own strategy. (21) is fundamentally a multi-constraint optimization

problem and thus can be tackled by again using the Lagrange multiplier method. To this end, the augmented cost function with respect to the driver's original cost function (15) shall be constructed. In doing so, a conflict over the driver's and the AFS controller's preview horizons N_1^p and N_2^p arises: from the driver's perspective, the augmented cost function needs to be evaluated over N_1^p stages whereas both (20a) and (20b) are evaluated over N_2^p stages. In light of this, N_1^p and N_2^p have to be set equal, that is, $N_1^p = N_2^p = N^p$. As a result, the driver's augmented cost function can be expressed as:

$$\begin{aligned}
J_1^{\text{LQDO}}(k) = & \frac{1}{2} \mathbf{w}^T(N^p | k) \mathbf{H}_1^T \mathbf{Q}_1 \mathbf{H}_1 \mathbf{w}(N^p | k) \\
& + \frac{1}{2} \sum_{j=0}^{N^p-1} [\mathbf{w}^T(j | k) \mathbf{H}_1^T \mathbf{Q}_1 \mathbf{H}_1 \mathbf{w}(j | k) + \delta_1(j | k)^2] \\
& + \frac{1}{2} \sum_{j=0}^{N^p-1} \{ \boldsymbol{\lambda}^T(j | k) [\mathbf{A}_w \mathbf{w}(j | k) + \mathbf{B}_{w1} \delta_1(j | k) \\
& \quad + \mathbf{B}_{w2} \delta_2(j | k) + \mathbf{F} \mathbf{R}^{\text{next}}(j | k) - \mathbf{w}(j+1 | k)] \} \quad (22) \\
& + \frac{1}{2} \sum_{j=0}^{N^p-1} \{ \boldsymbol{\mu}^T(j | k) [\mathbf{H}_2^T \mathbf{Q}_2 \mathbf{H}_2 \mathbf{w}(j | k) \\
& \quad + \mathbf{A}_w^T \mathbf{p}(j+1 | k) - \mathbf{p}(j | k)] \} \\
& + \frac{1}{2} \sum_{j=0}^{N^p-1} \{ \mathbf{v}^T(j | k) [\delta_2(j | k) + \mathbf{B}_{w2}^T \mathbf{p}(j+1 | k)] \}
\end{aligned}$$

where $\boldsymbol{\lambda}(j | k)$, $\boldsymbol{\mu}(j | k)$ and $\mathbf{v}(j | k)$ are respectively the Lagrange multipliers associated with the game progression equation, (20a), and (20b) for $j = 0, 1, 2, \dots, N^p - 1$. (22) reaches its minimum when the partial derivatives with respect to all its variables equal zero, that is

$$\frac{\partial J_1^{\text{LQDO}}(k)}{\partial \mathbf{w}(j | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N^p - 1, \text{ and } \frac{\partial J_1^{\text{LQDO}}(k)}{\partial \mathbf{w}(N^p | k)} = \mathbf{0} \quad (23a)$$

$$\frac{\partial J_2^{\text{LQDO}}(k)}{\partial \mathbf{p}(j+1 | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N^p - 1 \quad (23b)$$

$$\frac{\partial J_1^{\text{LQDO}}(k)}{\partial \delta_1(j | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N^p - 1 \quad (23c)$$

$$\frac{\partial J_1^{\text{LQDO}}(k)}{\partial \delta_2(j | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N^p - 1 \quad (23d)$$

$$\frac{\partial J_1^{\text{LQDO}}(k)}{\partial \boldsymbol{\mu}(j | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N^p - 1 \quad (23e)$$

$$\frac{\partial J_1^{\text{LQDO}}(k)}{\partial \mathbf{v}(j | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N^p - 1 \quad (23f)$$

$$\frac{\partial J_1^{\text{LQDO}}(k)}{\partial \boldsymbol{\lambda}(j | k)} = \mathbf{0} \text{ for } j = 0, 1, 2, \dots, N^p - 1 \quad (23g)$$

As a result, the necessary conditions for a minimum (22) are:

$$\begin{aligned}
\boldsymbol{\lambda}(j-1 | k) = & \mathbf{H}_1^T \mathbf{Q}_1 \mathbf{H}_1 \mathbf{w}(j | k) + \mathbf{A}_w^T \boldsymbol{\lambda}(j | k) + \mathbf{H}_2^T \mathbf{Q}_2 \mathbf{H}_2 \boldsymbol{\mu}(j | k) \\
\text{for } j = & 0, 1, 2, \dots, N^p - 1 \text{ with terminal condition} \\
\boldsymbol{\lambda}(N^p - 1 | k) = & \mathbf{H}_1^T \mathbf{Q}_1 \mathbf{H}_1 \mathbf{w}(N^p | k) \quad (24a)
\end{aligned}$$

$$\boldsymbol{\mu}(j+1 | k) = \mathbf{A}_w \boldsymbol{\mu}(j | k) + \mathbf{B}_w \mathbf{v}(j | k) \text{ for } j = 0, 1, 2, \dots, N^p - 2$$

with initial condition $\boldsymbol{\mu}(k) = \mathbf{0}$ and terminal condition

$$\boldsymbol{\mu}(N^p - 1 | k) + \mathbf{B}_w \mathbf{v}(N^p - 1 | k) = \mathbf{0} \quad (24b)$$

$$0 = \delta_1(j | k) + \mathbf{B}_{w1}^T \boldsymbol{\lambda}(j | k) \text{ for } j = 0, 1, 2, \dots, N^p - 1 \quad (24c)$$

$$0 = \mathbf{v}(j | k) + \mathbf{B}_{w2}^T \boldsymbol{\lambda}(j | k) \text{ for } j = 0, 1, 2, \dots, N^p - 1 \quad (24d)$$

Here (23a) to (23d) respectively lead to (24a) to (24d). (23e), (23f) and (23g) respectively lead to (20a), (20b) and (17).

Hungerländer and Neck [21] explained that (20a), (20b), (24a), (24b), (24c), and (24d) all together constitute a two-point boundary-value problem the solution to which results in the open-loop Stackelberg strategies of the players (driver and AFS controller in this case). To solve the problem, Hungerländer and Neck [21] supplemented two linear equations:

$$\begin{aligned}
\mathbf{p}(j+1 | k) = & \mathbf{M}^w(j+1 | k) \mathbf{w}(j+1 | k) \\
& + \mathbf{M}^\mu(j+1 | k) \boldsymbol{\mu}(j+1 | k) + \mathbf{m}(j+1 | k) \\
\text{for } j = & 0, 1, 2, \dots, N^p - 1 \quad (25)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\lambda}(j+1 | k) = & \mathbf{L}^w(j+1 | k) \mathbf{w}(j+1 | k) \\
& + \mathbf{L}^\mu(j+1 | k) \boldsymbol{\mu}(j+1 | k) + \mathbf{l}(j+1 | k) \\
\text{for } j = & 0, 1, 2, \dots, N^p - 1 \quad (26)
\end{aligned}$$

where $\mathbf{m}(j+1 | k)$, $\mathbf{l}(j+1 | k)$, $\mathbf{M}^w(j+1 | k)$, $\mathbf{M}^\mu(j+1 | k)$, $\mathbf{L}^w(j+1 | k)$, and $\mathbf{L}^\mu(j+1 | k)$ are matrices of appropriate dimensions. Due to space limitation, the intermediate algebraic steps are omitted whilst the driver and AFS steering control strategies across the game at time step k are presented. Details of the derivation can be found in [22]. It should be noted here that all the steps involved in the derivation are purely analytical-based; no numerical method is used.

$$\begin{aligned}
\delta_1^*(j | k) = & \mathbf{K}_1^{\text{LQDO}}(j | k) \{ \mathbf{x}(j | k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k) \}^T \\
\text{for } j = & 0, 1, 2, \dots, N^p - 1 \quad (27)
\end{aligned}$$

$$\begin{aligned}
\delta_2^*(j | k) = & \mathbf{K}_2^{\text{LQDO}}(j | k) \{ \mathbf{x}(j | k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k) \}^T \\
\text{for } j = & 0, 1, 2, \dots, N^p - 1 \quad (28)
\end{aligned}$$

(27) and (28) represent sequences of steering angles each of which corresponds to a particular game stage j . However, in reality only one steering angle can be applied to the vehicle at time step k . To this end, the “receding horizon” idea [23] is adopted involving taking the first element of each sequence, i.e. $\delta_1(0 | k)$ and $\delta_2(0 | k)$ as the steering angles to apply. Consequently, the driver and AFS theoretical Stackelberg steering control strategies can be respectively expressed as:

$$\delta_1^*(k) = \left[\mathbf{K}_1^{\text{LQDO-x}} \quad \mathbf{K}_1^{\text{LQDO-R}_1} \quad \mathbf{K}_1^{\text{LQDO-R}_2} \right] \begin{Bmatrix} \mathbf{x}(k) \\ \mathbf{R}_1(k) \\ \mathbf{R}_2(k) \end{Bmatrix} \quad (29)$$

$$\delta_2^*(k) = \left[\mathbf{K}_2^{\text{LQDO-x}} \quad \mathbf{K}_2^{\text{LQDO-R}_1} \quad \mathbf{K}_2^{\text{LQDO-R}_2} \right] \begin{Bmatrix} \mathbf{x}(k) \\ \mathbf{R}_1(k) \\ \mathbf{R}_2(k) \end{Bmatrix} \quad (30)$$

$\mathbf{K}_1^{\text{LQDO-x}}$ is the driver's gain array associated with vehicle state $\mathbf{x}(k)$. It is thereby named the driver's state gain array. $\mathbf{K}_1^{\text{LQDO-R}_1}$ is the gain array associated with the driver's previewed path $\mathbf{R}_1(k)$. It is thereby called the driver's preview gain array with $\mathbf{R}_1(k)$. Accordingly $\mathbf{K}_1^{\text{LQDO-R}_2}$ is called the driver's preview gain array with $\mathbf{R}_2(k)$. Similarly, $\mathbf{K}_2^{\text{LQDO-x}}$, $\mathbf{K}_2^{\text{LQDO-R}_1}$ and $\mathbf{K}_2^{\text{LQDO-R}_2}$ are respectively the AFS controller's state gain array, preview gain array with $\mathbf{R}_1(k)$, and that with $\mathbf{R}_2(k)$. These six gain arrays are all functions of matrices \mathbf{A} , \mathbf{B}_1 , \mathbf{B}_2 , \mathbf{Q}_1 and \mathbf{Q}_2 , and are thus time-invariant. Their determination can also be found in Appendix C in [22]. It can be seen that (29) and (30) agree in format with their conceptual expressions (5) and (6) developed in Section II.

B. Distributed Model Predictive Control (DMPC)

The application of Distributed Model Predictive Control (DMPC) approach to dynamic game problems was introduced by Rawlings and Mayne [24] in which derivation of Nash strategies was discussed. In this subsection, the procedure for taking the DMPC approach to the same driver-AFS interactive steering control problem as treated in the previous subsection is described. This involves four steps: construction of the AFS controller's prediction equation, derivation of the AFS optimal response, construction of the driver's prediction equation, and derivation of the driver's and the AFS controller's Stackelberg steering control strategies. It is noteworthy that the analysis presented below is, to the authors' knowledge, the first attempt of using the theory of Model Predictive Control (MPC) to addressing the equilibrium solution to a leader-follower game. The style of expression used by Maciejowski [23] is followed.

1) AFS Controller Prediction Equation

During the game played at time step k , vehicle orientations at game stages 1 and 2 can be predicted by iterating (7) as:

$$\begin{aligned} \mathbf{z}(1|k) &= \mathbf{C}\mathbf{x}(1|k) = \mathbf{C}[\mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(0|k) + \mathbf{B}_2\delta_2(0|k)] \\ \mathbf{z}(2|k) &= \mathbf{C}\mathbf{x}(2|k) = \mathbf{C}[\mathbf{A}^2\mathbf{x}(k) + \mathbf{B}_1\delta_1(1|k) + \mathbf{B}_2\delta_2(1|k) \\ &\quad + \mathbf{A}\mathbf{B}_1\delta_1(0|k) + \mathbf{A}\mathbf{B}_2\delta_2(0|k)] \end{aligned} \quad (31)$$

By continuing the iteration, the AFS controller could predict vehicle orientations up to N_2^p stages ahead. Hence, the AFS controller's (follower's) prediction equation at time step k is:

$$\mathbf{Z}_2(k) = \Psi_2\mathbf{x}(k) + \Theta_2\mathbf{U}_1(k) + \Omega_2\mathbf{U}_2(k) \quad (32)$$

where

$$\mathbf{Z}_2(k) = \begin{Bmatrix} \mathbf{z}(0|k) \\ \mathbf{z}(1|k) \\ \vdots \\ \mathbf{z}(N_2^p|k) \end{Bmatrix}, \quad \mathbf{U}_1(k) = \begin{Bmatrix} \delta_1(0|k) \\ \delta_1(1|k) \\ \vdots \\ \delta_1(N_2^u-1|k) \end{Bmatrix},$$

$$\mathbf{U}_2(k) = \begin{Bmatrix} \delta_2(0|k) \\ \delta_2(1|k) \\ \vdots \\ \delta_2(N_2^u-1|k) \end{Bmatrix}, \quad \Psi_2 = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{N_2^p} \end{bmatrix},$$

$$\Theta_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{CB}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{CA}^{N_2^p-1}\mathbf{B}_1 & \mathbf{CA}^{N_2^p-2}\mathbf{B}_1 & \cdots & \mathbf{CAB}_1 & \mathbf{CB}_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{CA}^{N_2^p-1}\mathbf{B}_1 & \mathbf{CA}^{N_2^p-2}\mathbf{B}_1 & \cdots & \mathbf{CA}^{N_2^p-N_2^u+1}\mathbf{B}_1 & \mathbf{C} \sum_{i=0}^{N_2^p-N_2^u} \mathbf{A}^i \mathbf{B}_1 \end{bmatrix},$$

and

$$\Omega_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{CB}_2 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{CA}^{N_2^p-1}\mathbf{B}_2 & \mathbf{CA}^{N_2^p-2}\mathbf{B}_2 & \cdots & \mathbf{CAB}_2 & \mathbf{CB}_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{CA}^{N_2^p-1}\mathbf{B}_2 & \mathbf{CA}^{N_2^p-2}\mathbf{B}_2 & \cdots & \mathbf{CA}^{N_2^p-N_2^u+1}\mathbf{B}_2 & \mathbf{C} \sum_{i=0}^{N_2^p-N_2^u} \mathbf{A}^i \mathbf{B}_2 \end{bmatrix}$$

In (32), N_2^u represents the AFS controller's control horizon which indicates the stages over which the AFS steering angle action varies. Since stage N_2^u the AFS steering action is held constant as $\delta_2(N_2^u-1|k)$ up to the preview horizon N_2^p . Therefore N_2^u must satisfy $N_2^u \leq N_2^p$.

2) AFS Controller Optimal Response

Given the AFS controller's prediction equation (32), the AFS controller's original cost function (13) can be rewritten as:

$$J_2^{\text{DMPC}}(k) = \|\mathbf{Z}_2(k) - \mathbf{R}_2(k)\|_{\mathbf{Q}_2^{\text{DMPC}}}^2 + \|\mathbf{U}_2(k)\|^2 \quad (33)$$

where $\|\mathbf{X}\|_{\mathbf{Y}}^2$ means $\mathbf{X}^T\mathbf{Y}\mathbf{X}$ and $\mathbf{Q}_2^{\text{DMPC}}$ is a diagonal matrix:

$$\mathbf{Q}_2^{\text{DMPC}} = \begin{bmatrix} \mathbf{Q}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}_2 \end{bmatrix}.$$

In (33) the AFS controller's control horizon N_2^u involved in $\mathbf{U}_2(k)$ is made equal to the AFS controller's preview horizon N_2^p to fit the formulation of the original cost function (13).

At this point, the AFS controller's "zero-input tracking error" $\boldsymbol{\varepsilon}_2(k)$ is defined according to Maciejowski's observation [23]:

$$\boldsymbol{\varepsilon}_2(k) = \mathbf{R}_2(k) - \Psi_2\mathbf{x}(k) - \Theta_2\mathbf{U}_1(k) \quad (34)$$

where $\boldsymbol{\varepsilon}_2(k)$ represents the difference between the AFS controller's target path $\mathbf{R}_2(k)$, and the predicted vehicle orientations $\mathbf{Z}_2(k)$ determined under the circumstance that the input $\mathbf{U}_2(k)$ is set zero. Substituting (34) into (33) then gives:

$$J_2^{\text{DMPC}}(k) = \|\Omega_2\mathbf{U}_2(k) - \boldsymbol{\varepsilon}_2(k)\|_{\mathbf{Q}_2^{\text{DMPC}}}^2 + \|\mathbf{U}_2(k)\|^2 \quad (35)$$

or equivalently,

$$J_2^{\text{DMPC}}(k) = \left\| \begin{bmatrix} \mathbf{S}_2 \{ \mathbf{\Omega}_2 \mathbf{U}_2(k) - \boldsymbol{\varepsilon}_2(k) \} \\ \mathbf{U}_2(k) \end{bmatrix} \right\|^2 \quad (36)$$

where \mathbf{S}_2 satisfies $\mathbf{S}_2^T \mathbf{S}_2 = \mathbf{Q}_2^{\text{DMPC}}$.

The AFS steering action sequence $\mathbf{U}_2(k)$ that minimizes cost function (36), or (13) must be the least-squares solution to

$$\begin{bmatrix} \mathbf{S}_2 \mathbf{\Omega}_2 \\ \mathbf{I} \end{bmatrix} \mathbf{U}_2(k) = \begin{bmatrix} \mathbf{S}_2 \\ \mathbf{0} \end{bmatrix} \boldsymbol{\varepsilon}_2(k) \quad (37)$$

where \mathbf{I} is an identity matrix of appropriate dimension.

Maciejowski [23] explained that a problem of this form can be solved by using the QR algorithm (invoked in Matlab using the backslash operator '\'). This leads to a numerical solution to $\mathbf{U}_2(k)$ which can be expressed as

$$\mathbf{U}_2(k) = \mathbf{K}_2^{\text{DMPC}} \boldsymbol{\varepsilon}_2(k) \quad (38)$$

where $\mathbf{K}_2^{\text{DMPC}}$ is a matrix resulting from the QR algorithm:

$$\mathbf{K}_2^{\text{DMPC}} = \begin{bmatrix} \mathbf{S}_2 \mathbf{\Omega}_2 \\ \mathbf{I} \end{bmatrix} \backslash \begin{bmatrix} \mathbf{S}_2 \\ \mathbf{0} \end{bmatrix} \quad (39)$$

$\mathbf{K}_2^{\text{DMPC}}$ is a function of \mathbf{A} , \mathbf{B}_2 , \mathbf{C} and \mathbf{Q}_2 , and is time-invariant.

Substituting (34) into (38) then gives:

$$\mathbf{U}_2(k) = \mathbf{K}_2^{\text{DMPC}} [-\boldsymbol{\Psi}_2 \mathbf{x}(k) + \mathbf{R}_2(k) - \boldsymbol{\Theta}_2 \mathbf{U}_1(k)] \quad (40)$$

(40) implies that the AFS controller's steering action sequence $\mathbf{U}_2(k)$ depends on vehicle state $\mathbf{x}(k)$, AFS target path $\mathbf{R}_2(k)$ and driver steering angle sequence $\mathbf{U}_1(k)$. In other words, (40) embodies that as the follower, the AFS controller responds to the driver's (leader's) action $\mathbf{U}_1(k)$ by adopting a specific strategy. Therefore, (40) is named the AFS controller's optimal response to any driver steering angle.

3) Driver Prediction Equation

The driver is assumed to make a preliminary estimation of vehicle orientations up to his or her preview horizon N_1^p at first. This can be accomplished by iterating vehicle dynamics equation (7) over N_1^p stages ahead, which leads to:

$$\mathbf{Z}_1(k) = \boldsymbol{\Psi}_1^{\text{pre}} \mathbf{x}(k) + \boldsymbol{\Theta}_1^{\text{pre}} \mathbf{U}_1(k) + \boldsymbol{\Omega}_1^{\text{pre}} \mathbf{U}_2(k) \quad (41)$$

where $\boldsymbol{\Psi}_1^{\text{pre}}$, $\boldsymbol{\Theta}_1^{\text{pre}}$ and $\boldsymbol{\Omega}_1^{\text{pre}}$ hold similar structures to $\boldsymbol{\Psi}_2$, $\boldsymbol{\Theta}_2$ and $\boldsymbol{\Omega}_2$ respectively, and both $\mathbf{U}_1(k)$ and $\mathbf{U}_2(k)$ are evaluated up to the driver's control horizon N_1^u . At this point, it should be noted that (41) does not represent the driver's prediction equation but just an intermediate equation.

The driver then tends to improve his or her preliminary estimation (41) by taking the AFS controller's optimal response (40) into account. This is achieved by substituting (40) into (41). It can be found that such substitution cannot be implemented unless the dimension of $\mathbf{U}_2(k)$ in (40), i.e. $N_2^p \times 1$ (N_2^u has been made to N_2^p for constructing (33)) equals that of $\mathbf{U}_2(k)$ in (41), i.e. $N_1^u \times 1$. To this end, $N_1^u = N_2^u = N_2^p = N^p$ is assumed. As a result, the substitution yields:

$$\mathbf{Z}_1(k) = \boldsymbol{\Psi}_1 \mathbf{x}(k) + \boldsymbol{\Theta}_1 \mathbf{U}_1(k) + \boldsymbol{\Omega}_1 \mathbf{R}_2(k) \quad (42)$$

where $\boldsymbol{\Psi}_1 = \boldsymbol{\Psi}_1^{\text{pre}} - \boldsymbol{\Omega}_1^{\text{pre}} \mathbf{K}_2^{\text{DMPC}} \boldsymbol{\Psi}_2$, $\boldsymbol{\Theta}_1 = \boldsymbol{\Theta}_1^{\text{pre}} - \boldsymbol{\Omega}_1^{\text{pre}} \mathbf{K}_2^{\text{DMPC}} \boldsymbol{\Theta}_2$, and $\boldsymbol{\Omega}_1 = \boldsymbol{\Omega}_1^{\text{pre}} \mathbf{K}_2^{\text{DMPC}}$. (42) is the driver's prediction equation.

4) Theoretical Stackelberg Strategies

Given the driver's prediction equation (42), his or her cost function (11) can be then rewritten as:

$$J_1^{\text{DMPC}}(k) = \|\mathbf{Z}_1(k) - \mathbf{R}_1(k)\|_{\mathbf{Q}_1^{\text{DMPC}}}^2 + \|\mathbf{U}_1(k)\|^2 \quad (43)$$

where $\mathbf{Q}_1^{\text{DMPC}}$ is a diagonal matrix:

$$\mathbf{Q}_1^{\text{DMPC}} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}_1 \end{bmatrix},$$

and the driver's control horizon N_1^u involved in $\mathbf{U}_1(k)$ needs to be made equal to the driver's preview horizon N_1^p to fit the formulation of the driver's original cost function (11). Accordingly, $N_1^u = N_2^u = N_1^p = N_2^p = N^p$ holds hereinafter.

Grounded in (42), the driver's "zero-input tracking error" $\boldsymbol{\varepsilon}_1(k)$ can be defined in a similar way to that for $\boldsymbol{\varepsilon}_2(k)$

$$\boldsymbol{\varepsilon}_1(k) = \mathbf{R}_1(k) - \boldsymbol{\Psi}_1 \mathbf{x}(k) - \boldsymbol{\Omega}_1 \mathbf{R}_2(k) \quad (44)$$

On this basis, the driver steering angle sequence $\mathbf{U}_1(k)$ that minimizes cost function (43), or equivalently (11) can be described as the least-squares solution to

$$\begin{bmatrix} \mathbf{S}_1 \boldsymbol{\Theta}_1 \\ \mathbf{I} \end{bmatrix} \mathbf{U}_1(k) = \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{0} \end{bmatrix} \boldsymbol{\varepsilon}_1(k) \quad (45)$$

where \mathbf{S}_1 satisfies $\mathbf{Q}_1^{\text{DMPC}} = \mathbf{S}_1^T \mathbf{S}_1$ and \mathbf{I} is an identity matrix of appropriate dimension.

Hence, $\mathbf{U}_1(k)$ can be determined as

$$\mathbf{U}_1(k) = \mathbf{K}_1^{\text{DMPC}} \boldsymbol{\varepsilon}_1(k) \quad (46)$$

where $\mathbf{K}_1^{\text{DMPC}}$ is calculated through the QR algorithm as

$$\mathbf{K}_1^{\text{DMPC}} = \begin{bmatrix} \mathbf{S}_1 \boldsymbol{\Theta}_1 \\ \mathbf{I} \end{bmatrix} \backslash \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{0} \end{bmatrix} \quad (47)$$

It can be seen from (47) that $\mathbf{K}_1^{\text{DMPC}}$ is a function of \mathbf{A} , \mathbf{B}_1 , \mathbf{B}_2 , \mathbf{C} , \mathbf{Q}_1 and \mathbf{Q}_2 , and is time-invariant.

Substituting (44) into (46) then gives:

$$\mathbf{U}_1(k) = \mathbf{K}_1^{\text{DMPC}} [\mathbf{R}_1(k) - \boldsymbol{\Psi}_1 \mathbf{x}(k) - \boldsymbol{\Omega}_1 \mathbf{R}_2(k)] \quad (48)$$

Here $\mathbf{U}_1(k)$ is a vector of N^p elements comprising $\delta_1(0|k)$ to $\delta_1(N^p|k)$. These elements denote respectively the driver's optimal steering wheel angles determined at each stage from 0 up to N^p across the game played at time step k . However, as explained previously, in reality the driver can only apply one particular steering angle at time step k . In view of this, the "receding horizon" idea [23] is used again and the first element in $\mathbf{U}_1(k)$, i.e. $\delta_1(0|k)$ is chosen as the steering wheel angle

applied at time step k . Therefore the driver's theoretical Stackelberg steering control strategy can be expressed as:

$$\delta_1^*(k) = \begin{bmatrix} \mathbf{K}_1^{\text{DMPC}-\mathbf{x}} & \mathbf{K}_1^{\text{DMPC}-\mathbf{R}_1} & \mathbf{K}_1^{\text{DMPC}-\mathbf{R}_2} \end{bmatrix} \begin{Bmatrix} \mathbf{x}(k) \\ \mathbf{R}_1(k) \\ \mathbf{R}_2(k) \end{Bmatrix} \quad (49)$$

where $\mathbf{K}_1^{\text{DMPC}-\mathbf{x}}$, $\mathbf{K}_1^{\text{DMPC}-\mathbf{R}_1}$ and $\mathbf{K}_1^{\text{DMPC}-\mathbf{R}_2}$ respectively denote the driver's state gain array, preview gain array with $\mathbf{R}_1(k)$, and preview gain array with $\mathbf{R}_2(k)$. These three gain arrays are all functions of matrices \mathbf{A} , \mathbf{B}_1 , \mathbf{B}_2 , \mathbf{C} , \mathbf{Q}_1 and \mathbf{Q}_2 and are thus time-invariant. Specifically, $\mathbf{K}_1^{\text{DMPC}-\mathbf{x}}$ is determined as the first row of $-\mathbf{K}_1^{\text{DMPC}} \Psi_1$, $\mathbf{K}_1^{\text{DMPC}-\mathbf{R}_1}$ is the first row of $\mathbf{K}_1^{\text{DMPC}}$, and $\mathbf{K}_1^{\text{DMPC}-\mathbf{R}_2}$ is the first row of $-\mathbf{K}_1^{\text{DMPC}} \Omega_1$.

By substituting (48) into (40), and applying the receding horizon approach, the AFS controller's theoretical Stackelberg steering control strategy can then be achieved:

$$\delta_2^*(k) = \begin{bmatrix} \mathbf{K}_2^{\text{DMPC}-\mathbf{x}} & \mathbf{K}_2^{\text{DMPC}-\mathbf{R}_1} & \mathbf{K}_2^{\text{DMPC}-\mathbf{R}_2} \end{bmatrix} \begin{Bmatrix} \mathbf{x}(k) \\ \mathbf{R}_1(k) \\ \mathbf{R}_2(k) \end{Bmatrix} \quad (50)$$

where $\mathbf{K}_2^{\text{DMPC}-\mathbf{x}}$, $\mathbf{K}_2^{\text{DMPC}-\mathbf{R}_1}$ and $\mathbf{K}_2^{\text{DMPC}-\mathbf{R}_2}$ have similar format to $\mathbf{K}_1^{\text{DMPC}-\mathbf{x}}$, $\mathbf{K}_1^{\text{DMPC}-\mathbf{R}_1}$ and $\mathbf{K}_1^{\text{DMPC}-\mathbf{R}_2}$, respectively.

By comparing (49) and (50) respectively to (29) and (30), it can be seen that the DMPC and the LQDO approaches give comparable expressions of open-loop theoretical Stackelberg steering control strategies. However, the two approaches differ in the algebraic procedures for strategy derivation. Specifically, the LQDO approach develops the Stackelberg strategies by identifying and solving a two-point boundary-value problem. Each algebraic step involved in the solution procedure is derived analytically. On the other hand, the DMPC approach develops the Stackelberg strategies by solving least-squares problems, during which the QR algorithm is used and numerical solutions are yielded. In the next section, numerical results obtained using these two approaches are compared.

IV. NUMERICAL RESULTS

In this section, studies towards the LQDO and the DMPC approaches are performed via simulation. Two types of numerical results are compared: 1) gain arrays of the driver and the AFS controller, and 2) computation time.

A. Gain Array Comparison

The driver's and the AFS controllers' gain arrays resulting from the LQDO approach, i.e. the gain arrays appearing in (29) and (30) are compared with those generated using the DMPC one, i.e. the gain arrays in (49) and (50). Such a comparison is conducted to examine whether the two approaches give similar numerical results in terms of driver and AFS control actions. The vehicle parameters used are measured from an Opel Signum car, and are exactly the same as those listed in [12]. It should be noted that the dynamics of the AFS actuation system is ignored and the ratio of AFS steering motor angle to vehicle

TABLE I
COMPARISON OF CPU TIME CONSUMPTION

Game stage N^p	LQDO CPU time (s)	DMPC CPU time (s)
50	9.80	0.10
100	141.45	0.16
200	2199.77	0.66

road wheel angles is set equal to that between driver steering wheel and road wheels. On this basis, the following driver and AFS parameters are used: $q_1^y = 3.6e-2$, $q_1^w = 2.0e-2$, $q_2^y = 2.5e-2$, $q_2^w = 1.0e-2$ and $N^p = 200$. These values may not represent realistic behaviour of a human driver and an AFS controller but are employed simply for gain array comparison.

Comparison of the driver's state gain arrays, i.e., of $\mathbf{K}_1^{\text{LQDO}-\mathbf{x}}$ and $\mathbf{K}_1^{\text{DMPC}-\mathbf{x}}$ is presented in Fig. 3 (a). Comparison of the AFS controller's state gain arrays $\mathbf{K}_2^{\text{LQDO}-\mathbf{x}}$ and $\mathbf{K}_2^{\text{DMPC}-\mathbf{x}}$ is shown in Fig. 3 (b). Comparison of the driver's preview gain arrays with $\mathbf{R}_1(k)$, i.e. of $\mathbf{K}_1^{\text{LQDO}-\mathbf{R}_1}$ and $\mathbf{K}_1^{\text{DMPC}-\mathbf{R}_1}$ is presented in Fig. 4 (a) while that of the AFS corresponding gain arrays $\mathbf{K}_2^{\text{LQDO}-\mathbf{R}_1}$ and $\mathbf{K}_2^{\text{DMPC}-\mathbf{R}_1}$ is shown in Fig. 4 (b). Finally, comparison of the driver's preview gain arrays with $\mathbf{R}_2(k)$, i.e. of $\mathbf{K}_1^{\text{LQDO}-\mathbf{R}_2}$ and $\mathbf{K}_1^{\text{DMPC}-\mathbf{R}_2}$ is presented in Fig. 5(a) while that of the AFS gain arrays $\mathbf{K}_2^{\text{LQDO}-\mathbf{R}_2}$ and $\mathbf{K}_2^{\text{DMPC}-\mathbf{R}_2}$ is shown in Fig. 5 (b). It can be seen that the LQDO and the DMPC yield visually identical gain arrays. Besides the driver and AFS controller values used above, more values have been tested and identical results were always obtained between the two approaches. This suggests that the LQDO and the DMPC are equivalent in representing driver and AFS open-loop Stackelberg steering control strategies. It is noteworthy that it is impossible to have a comparison between the analytical expressions of Stackelberg strategies derived from the two approaches. The reason is that in the DMPC approach intrinsically involves the use of QR algorithm which results in numerical solutions, as described by (39) and (47).

B. Computation Time Comparison

The computation times consumed by the two approaches are compared in three cases where the game stage N^p is set respectively 50, 100 and 200. Values of all the other parameters are kept the same as those used in the previous subsection. The computer used has a 2.40 GHz CPU and an 8.00 GB RAM. The recorded CPU times are presented in Table I. It can be seen that in comparison to the LQDO approach, the DMPC one consumes far less CPU time in all cases. Besides, as N^p increases, the CPU time cost by the LQDO approach increases severely. This is because the $\mathbf{K}_1^{\text{LQDO}}(0|k)$ and $\mathbf{K}_2^{\text{LQDO}}(0|k)$ required in the LQDO approach are derived analytically through a backward iteration process, starting from game stage $N^p - 1$ back to 0 [22]. Such iteration is computationally expensive, especially when N^p is relatively large, as those assumed in the present study. By contrast, the $\mathbf{K}_1^{\text{DMPC}}$ and $\mathbf{K}_2^{\text{DMPC}}$ required in the DMPC approach are determined numerically via the QR algorithm which does not involve time-consuming iteration and is thus more efficient.

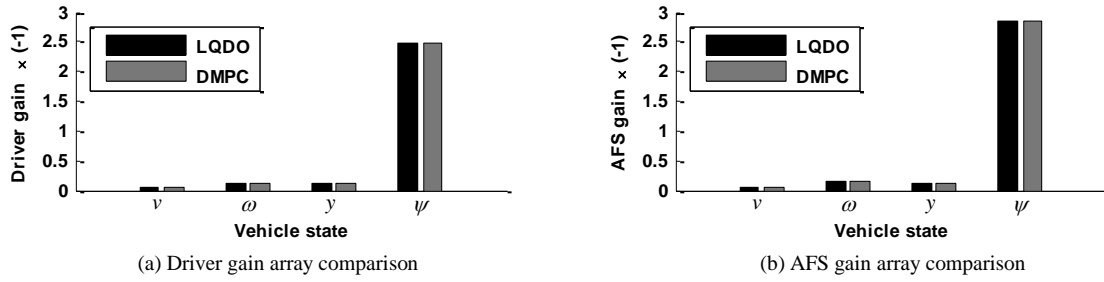


Fig. 3. Comparison of state gain arrays

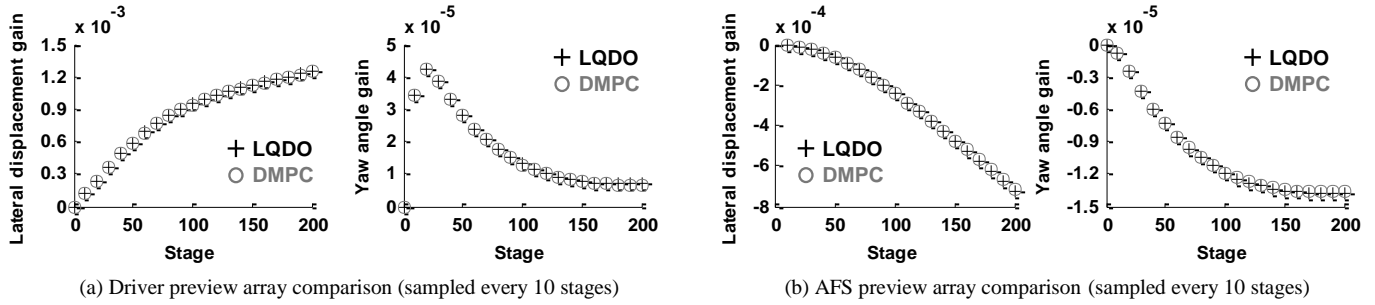
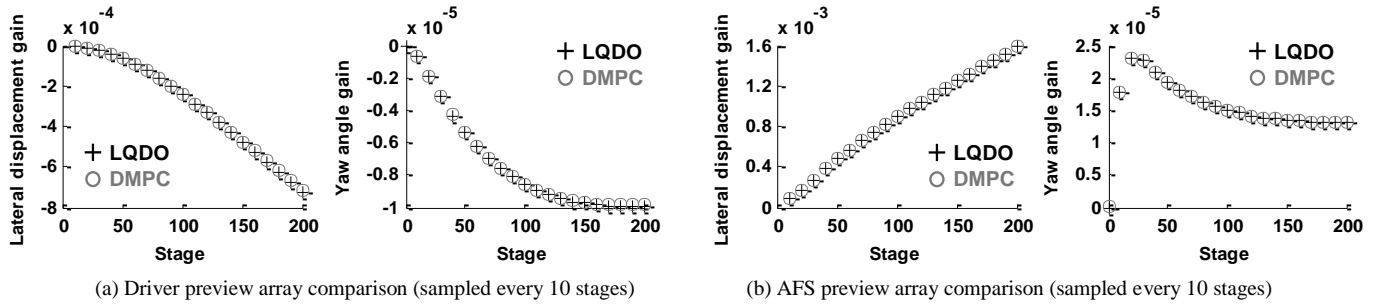
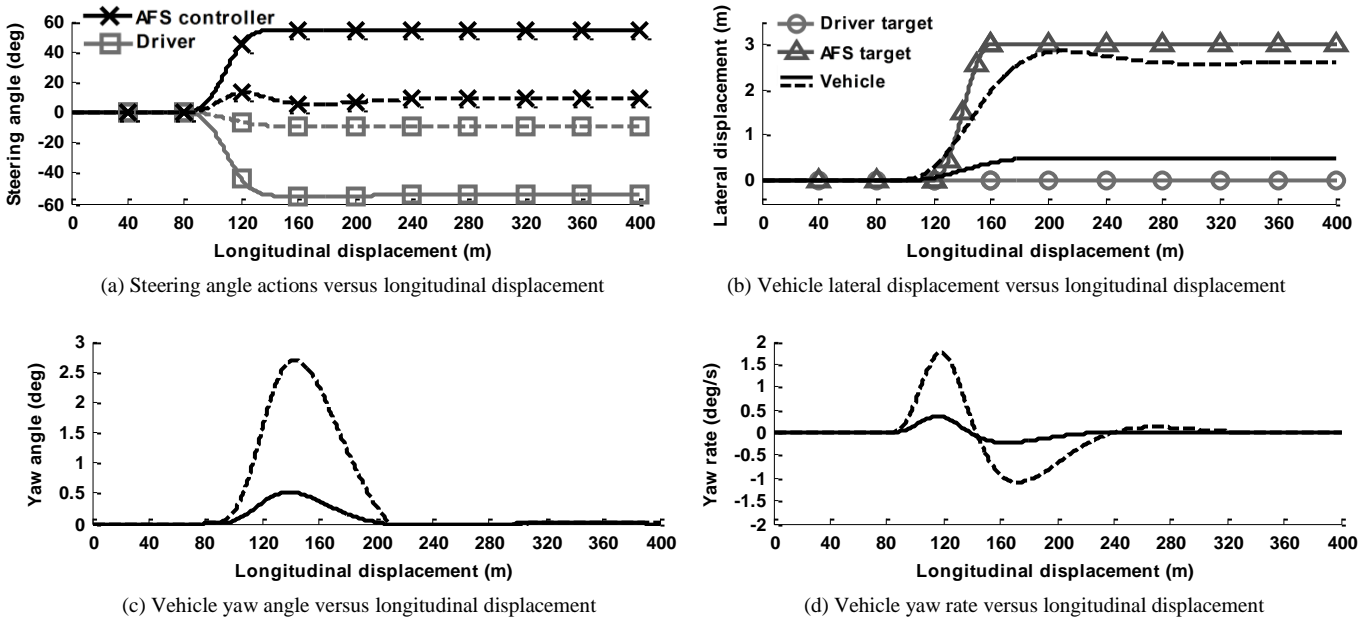
Fig. 4. Comparison of preview gain arrays with R_1 Fig. 5. Comparison of preview gain arrays with R_2 

Fig. 6. Time histories of steering control actions and vehicle lateral response. Solid: driver parameter set 1); dashed: driver parameter set 2)

V. DERIVATION OF PRACTICAL STACKELBERG STEERING CONTROL STRATEGIES

The driver-AFS interactive steering control outlined in Fig. 2 suggests a possible way of modelling a driver's interaction with a vehicle AFS system compensating for the driver's steering angle. Validation of the driver model using measured driver behaviour is needed. In preparation for such work, the steering control strategy of the AFS controller should be implemented in a driving simulator or instrumented vehicle. However, it would be impractical to use the AFS theoretical Stackelberg steering control strategy (30), or equivalently (50) derived in Section III. The reason is that it is technically difficult for the AFS to identify the driver's previewed target path $\mathbf{R}_1(k)$. To solve this difficulty, modifications are introduced to the development of the AFS open-loop Stackelberg steering control strategy in the DMPC framework. This results in a "practical" Stackelberg strategy for the AFS, which acts as the counterpart of the "theoretical" strategy (30), or equivalently (50). Accordingly, the driver's "theoretical" Stackelberg strategy (29), or equivalently (49) is modified. On this basis, time histories of driver and AFS control actions and vehicle response are simulated in an obstacle avoidance scenario.

A. Practical Stackelberg Strategies

It was described in Section III that in the DMPC context control horizon can be set different from preview horizon. In view of this, modifications are applied to both the driver and the AFS by reducing their respective control horizons to unity, i.e. $N_1^u = N_2^u = 1$. Controllers with such unity control horizon have been utilized by Ungoren and Peng [25] for driver modelling and by Keviczky *et al.* [26] for vehicle steering. As a result, the AFS controller's prediction equation (32) becomes:

$$\mathbf{Z}_2^{\text{prac}}(k) = \Psi_2^{\text{prac}} \mathbf{x}(k) + \Theta_2^{\text{prac}} \delta_1(k) + \Omega_2^{\text{prac}} \delta_2(k) \quad (51)$$

where $\mathbf{Z}_2^{\text{prac}}(k)$ and Ψ_2^{prac} hold the same structure as $\mathbf{Z}_2(k)$ and Ψ_2 in (32) whilst Θ_2^{prac} and Ω_2^{prac} are now column vectors as per Θ_2 and Ω_2 in (32). Based on (51), the AFS practical Stackelberg steering control strategy (52) can be derived by following the same procedure as presented from (32) to (40):

$$\delta_2(k) = \mathbf{K}_2^{\text{prac}} [-\Psi_2^{\text{prac}} \mathbf{x}(k) + \mathbf{R}_2(k) - \Theta_2^{\text{prac}} \delta_1(k)] \quad (52)$$

(52) suggests that the AFS controller's steering action $\delta_2(k)$ at time step k can be determined as a function of AFS target path $\mathbf{R}_2(k)$, driver's steering wheel angle $\delta_1(k)$ and vehicle state $\mathbf{x}(k)$. Since the rationality of assuming available $\mathbf{x}(k)$, $\mathbf{R}_2(k)$ and $\delta_1(k)$ was declared in Section II, (52) can be implemented in an AFS system in a test vehicle.

Comparable modifications are then applied to the driver's prediction equation (42). This leads to:

$$\mathbf{Z}_1^{\text{prac}}(k) = \Psi_1^{\text{prac}} \mathbf{x}(k) + \Theta_1^{\text{prac}} \delta_1(k) + \Omega_1^{\text{prac}} \mathbf{R}_2(k) \quad (53)$$

By following the procedure from (41) to (48), the driver's practical Stackelberg steering control strategy can be derived:

$$\delta_1(k) = \mathbf{K}_1^{\text{prac}} [\mathbf{R}_1(k) - \Psi_1^{\text{prac}} \mathbf{x}(k) - \Omega_1^{\text{prac}} \mathbf{R}_2(k)] \quad (54)$$

At this point, human drivers' steering control behaviour in response to the AFS practical Stackelberg steering control strategy (52) can be then measured through carefully designed experiments. The driver Stackelberg steering control strategy (54) can be then used to fit the measured driver behaviour so as to examine its validity. It should be noted that the modifications described above cannot be applied in the LQDO framework since the LQDO does not allow the control horizon to be set independent of the preview horizon. In the next subsection, simulated time histories of the driver and AFS steering angles and vehicle responses in an obstacle avoidance scenario will be displayed. Driver behaviour measurements or model validation is not presented but reserved for a future paper.

B. Time Histories

An obstacle avoidance scenario is designed for simulation study. Specifically, the AFS controller is assumed to have detected an obstacle ahead and is about to execute a lane change for collision avoidance. However, the driver is assumed to keep the vehicle travelling straight ahead. Such a conflict may occur when the driver believes that his or her control will not cause a collision to the obstacle, e.g. a pedestrian dashing across the road, or the driver thinks that his or her control will result in an outcome which is more ethically acceptable, e.g. collision with a car rather than a pedestrian, as illustrated by Gerdes and Thornton [27]. The vehicle parameters are set the same as those used in Section VI. The AFS practical strategy (52) is set to adopt $q_2^y = 6e - 4$, $q_2^w = 0$ and $N_2^p = 200$. It was reported in [22] that such a set gave good path-following accuracy whilst offered drivers the opportunity to override the AFS controller. With regard to the driver model, two sets of values are used: 1) ($q_1^y = 1e - 2$, $q_1^w = 0$, $N_1^p = 200$) which represents a driver who applies stronger steering control than the AFS controller, and 2) ($q_1^y = 3e - 4$, $q_1^w = 0$, $N_1^p = 200$) which represents a driver who exerts weaker control. Fig. 6 (a) displays the driver and AFS steering angles over vehicle longitudinal displacement. Fig. 6 (b) shows the driver and AFS target paths and the vehicle lateral displacement. Fig. 6 (c) and (d) respectively exhibits vehicle yaw angle and yaw rate. Since the vehicle is modelled to travel at a constant speed of 20 m/s, the two subplots are essentially time histories. It can be seen that the driver and the AFS gave steering angles in opposite direction as they tried to follow their individual target paths. For the driver parameter set 1) (solid lines), the vehicle tracked a straight line much closer to the driver's target path. For set 2) (dashed lines), the vehicle is predominantly controlled by the AFS to track its lane change target path. Consequently, the vehicle yaw responses as shown in Fig. 6 (c) and (d) are more severe than that of set 1). Fig. 6 also suggests that by altering the driver's path-error weights q_1^y and q_1^w , various driver steering control behaviour can be yielded, varying from the situation where the vehicle path following is dominated by the driver (solid lines) to that by the AFS controller (dashed lines).

VI. CONCLUSION

This paper concerns the adoption of Stackelberg equilibrium to modelling a human driver's interaction with a vehicle Active Front Steering (AFS) controller in a dynamic game framework. The driver and the AFS controller's open-loop steering control strategies are derived firstly following a procedure reported in literature based on the Linear Quadratic Dynamic Optimization (LQDO) approach and then another procedure introduced by the authors based on the Distributed Model Predictive Control (DMPC) approach. The following conclusions were obtained:

The LQDO approach derives the theoretical Stackelberg strategies by identifying and solving a two-point boundary-value problem. Each algebraic step involved in the derivation is on a purely analytical basis. On the other hand, the DMPC approach derives the Stackelberg strategies by identifying a least-squares problem and solving it using the QR algorithm. This leads to numerical solutions. Such differences suggest that comparison of analytical expressions of theoretical Stackelberg strategies derived from the two approaches cannot be made.

The two approaches were compared via a simulation and were found to yield visually identical control gain arrays. This suggests that these two approaches are equivalent in terms of representing driver and AFS steering control behaviour.

The DMPC approach used much less CPU times than the LQDO one in simulating driver and AFS control gains.

A modification to the AFS theoretical Stackelberg steering control strategy was made in the DMPC framework involving reducing the AFS controller's control horizon to unity. Such a modification resulted in the AFS controller's practical Stackelberg steering control strategy that can be implemented in a driving simulator or a test vehicle for driver behaviour measurement. This modification cannot be made in the LQDO framework due to that the LQDO approach does not in theory accept a "control horizon" independent of a "preview horizon".

A case study was performed to illustrate how a driver might interact with a vehicle AFS controller in an obstacle avoidance scenario. Simulation results indicated that by altering the driver path-error weights, various driver steering control behaviour can be produced by using the driver model developed.

REFERENCES

- [1] W. Klier, G. Relmann, and W. Reinelt, "Concept and functionality of the Active Front Steering system," Society of Automotive Engineers, Warrendale, PA, USA (SAE Technical Paper 2004-21-0073), Oct. 2004.
- [2] R. Isermann, R. Mannale, and K. Schmitt, "Collision-avoidance systems PRORETA: situation analysis and intervention control," *J. Control. Eng. Pract.*, vol. 20, no. 11, pp. 1236-1246, Nov. 2012.
- [3] S. J. Anderson, J. M. Walker, and K. Iagnemma, "Experimental performance analysis of a homotopy-based shared autonomy framework," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 2, pp. 190-199, Apr. 2014.
- [4] M. Plöchl and J. Edelmann, "Driver models in automobile dynamics application," *Veh. Syst. Dyn.*, vol. 45, no. 7-8, pp. 699-741, Jun. 2007.
- [5] R. B. Myerson, *Games Theory: Analysis of Conflict*. Cambridge, MA, USA: Harvard Univ. Press, 1991, ch. 1.
- [6] D. A. Braun, P. A. Ortega, and D. M. Wolpert, "Nash equilibria in multi-agent motor interactions," *PLoS Comput. Biology*, vol. 5, no. 8, pp. 1-8, Aug. 2009.
- [7] E. J. Dockner and R. Neck, "Time consistency, subgame perfectness, solution concepts and information patterns in dynamic models of stabilization policies," in *Adv. Comput. Econ.*, vol. 22, pp. 51-101, 2008.

- [8] I. K. Geçkil and P. L. Anderson. *Applied Game Theory and Strategic Behavior*. Boca Raton, FL, USA: CRC Press, 2010, ch. 2.
- [9] C. I. Chen and J. B. Cruz, "Stackelberg solution for two-person games with biased information patterns," *IEEE Trans. Autom. Control*, vol. 17, no. 6, pp. 791-798, Dec. 1972.
- [10] T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory*, 2nd ed. New York, NY, USA: Academic, 1995.
- [11] S. H. Tamaddoni, S. Taheri, and M. Ahmadian, "Optimal preview game theory approach to vehicle stability controller design," *Veh. Syst. Dyn.*, vol. 49, no. 12, pp. 1967-1979, Dec. 2011.
- [12] X. Na and D. J. Cole, "Game-theoretic modeling of the steering interaction between a human driver and a vehicle collision avoidance controller," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 1, pp. 25-38, Feb. 2015.
- [13] C. Dextreit and I. V. Kolmanovsky, "Game theory controller for hybrid electric vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 652-663, Mar. 2014.
- [14] H. Chen, J. Kessels, M. Donkers, and S. Weiland, "Game-theoretic approach for complete vehicle energy management," presented at *IEEE Veh. Power and Propulsion Conf.*, Coimbra, Portugal, 2014.
- [15] X. Na, and D. J. Cole, "Investigation of a driver's interaction with an active steering collision avoidance system," presented at the *23rd Int. Symp. Dynamics Vehicles on Roads Tracks*, Qingdao, China, 2013.
- [16] D. J. Cole, A. J. Pick, and A. M. C. Odhams, "Predictive and linear quadratic methods for potential application to modelling driver steering control," *Veh. Syst. Dyn.*, vol. 44, no. 3, pp. 259-284, Mar. 2006.
- [17] J. Engwerda, *LQ Dynamic Optimization and Differential Games*. Sussex, U.K.: Wiley, 2005, ch. 1.
- [18] R. S. Sharp and V. Valtetsiotis, "Optimal preview car steering control," *Veh. Syst. Dyn.*, vol. 35, suppl., pp. 101-117, Dec. 2001.
- [19] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. Hoboken, NJ, USA: Wiley, 2012, ch. 2.
- [20] D. Soudbakhsh and A. Eskandarian, "Steering control collision avoidance system and verification through subject study," *IET Intell. Transp. Syst.*, vol. 9, no. 10, pp. 907-915, 2015.
- [21] P. Hungerländer and R. Neck, "An algorithmic equilibrium solution for n-person dynamic Stackelberg difference games with open-loop information pattern," *Dynamic Modeling and Econometric in Economics and Finance*, vol. 13, pp. 197-214, 2011.
- [22] X. Na and D. J. Cole, "Modelling Driver-AFS interactive steering control using the principle of open-loop Stackelberg equilibrium," Dept. Eng., Univ. Cambridge, Cambridge, UK, Tech. Rep. CUED/TR. 101, 2013.
- [23] J. M. Maciejowski, *Predictive Control: With Constraints*. London, U.K.: Prentice-Hall, 2002, ch. 3.
- [24] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. New York: NY, USA: Nob Hill, 2009, ch. 6.
- [25] A. Y. Ungoren and H. Peng, "An adaptive lateral preview driver model," *Veh. Syst. Dyn.*, vol. 43, no. 4, pp. 245-259, Apr. 2005.
- [26] T. Keviczky, P. Falcone, F. Borrelli, J. Asgari and D. Hrovat, "Predictive control approach to autonomous vehicle steering," presented at the *American Control Conf.*, Minneapolis, USA, 2006.
- [27] J.C. Gerdes and S. M. Thorton, "Implementable ethics for autonomous vehicles," in *Autonomes Fahren*. Berlin, Germany: Springer-Verlag, 2015, ch. 2.



Xiaoxiang Na received the B.Sc. and M.Sc. degrees in automotive engineering from the College of Automotive Engineering, Jilin University, China, in 2007 and 2009, respectively. He received the Ph.D. degree in driver-vehicle dynamics from the Department of Engineering, University of Cambridge (CUED), U.K. in 2014, where he is currently a Research Associate.

His main research interests include driver-vehicle dynamics and vehicle in-service monitoring



David J. Cole received the B.A. degree in engineering and the Ph.D. degree in vehicle dynamics from the Department of Engineering, University of Cambridge (CUED), U.K. in 1985 and 1990, respectively.

From 1990 to 1996, he undertook his postdoctoral research in heavy vehicle dynamics at CUED, where he is currently a Senior Lecturer. From 1996 to 2000, he was a Lecturer with the University of Nottingham, U.K.

In 2000, he returned to CUED to take up his present position. His main research interest include driver-vehicle dynamics.