View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Apollo

Game Theoretic Modelling of a Human Driver's Steering Interaction with Vehicle Active Steering Collision Avoidance System

Xiaoxiang Na* and David J. Cole

Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK

*Corresponding author.

Xiaoxiang Na is with the Department of Engineering, University of Cambridge, CB2 1PZ, Cambridge, U.K. (e-mail: xnhn2@cam.ac.uk).

David J. Cole is with the Department of Engineering, University of Cambridge, CB2 1PZ, Cambridge, U.K. (e-mail: djc13@cam.ac.uk).

***Abstract*—**Development of vehicle active steering collision avoidance systems calls for mathematical models capable of predicting a human driver's response so as to reduce the cost involved in field tests whilst accelerate product development. This article provides a discussion on the paradigms that may be used for modelling a driver's steering interaction with vehicle collision avoidance control in path-following scenarios. Four paradigms, namely decentralized, noncooperative Nash, noncooperative Stackelberg and cooperative Pareto are established. The decentralized paradigm, developed based on optimal control theory, represents a driver's interaction with the collision avoidance controllers that disregard driver steering control. The noncooperative Nash and Stackelberg paradigms are used for predicting a driver's steering behaviour in response to the collision avoidance control that actively compensates for driver steering action. These two are devised based on the principles of equilibria in noncooperative game theory. The cooperative Pareto paradigm is derived from cooperative game theory to model a driver's interaction with the collision avoidance systems that take into account the driver's target path. The driver and the collision avoidance controllers' optimization problems and their resulting steering strategies arise in each paradigm are delineated. Two mathematical approaches applicable to these optimization problems, namely the distributed Model Predictive Control and the Linear Quadratic dynamic optimization approaches are described in some detail. A case study illustrating a conflict in steering control between driver and vehicle collision avoidance system is performed via simulation. It was found that variation of driver path-error cost function weights results in a variety of steering behaviours which are distinct between paradigms.

***Index Terms*—**Driver, Vehicle, Active Steering Collision Avoidance, Interaction, Modelling, Game Theory

I. INTRODUCTION

ACTIVE FRONT STEERING (AFS) technology enables an angle to be superimposed upon the steering angle generated by a human driver. It allows vehicle handling and stability to be improved [1]. Recently, AFS was used for automatic collision avoidance control [2]-[4]. Such control generally involves three steps: surrounding objects and road boundaries are detected by onboard sensing devices; a target path for collision avoidance is planned by AFS controller when an imminent collision is detected; a steering angle action is finally applied by AFS controller to guide vehicle to follow the target path.

Development of vehicle active-steering-based collision avoidance control currently relies largely on experimental approaches using test drivers [5]-[7]. A consequence is that the process is time-consuming and expensive. Alternatively, decision-making could be supported in low-cost design phases using mathematical models capable of reproducing drivers' reactions to vehicle active steering intervention. Such models may also allow deeper insights into physiological and cognitive behaviours of human drivers so that optimization of present or future driver-automation interfaces, e.g. continuous sharing control [8] becomes a possibility. However, little attention has yet been paid to this particular research arena. One example of the limited work is by Cole [9] who proposed a driver model with arm neuromuscular dynamics to investigate driver response to a step angle fault in AFS system. The aim of the work described in the present paper is to address the lack of theoretical analysis of the interaction between driver steering and vehicle AFS collision avoidance control.

In a collision avoidance scenario, when the AFS controller determines its steering control using merely its planned target path and vehicle state feedback but not taking the driver's handwheel action into consideration, such as that described in [3], the driver may react by ignoring the AFS steering action in return whilst deriving his/her steering wheel control using the driver's planned target path and perceived vehicle states. Such driver behaviour can be modelled using optimal control theory [10] where the driver and the AFS controller are treated as independent authorities. On the other hand, when the AFS controller is designed to neutralize the effects of the driver's steering angle action [2], the driver may respond by further increasing

his/her steering wheel angle input to compensate for the effects of AFS control. In this event, each of the two controllers appears to take into account the other's control in the development of its own steering action. Such driver-AFS interaction corresponds to the features of a dynamic game of which the definitions have been detailed by Cruz [11], and Basar and Olsder [12]. Under such circumstance, dynamic game theory is used in the present article to shape the steering interplay between the driver and the AFS controller.

There are three aspects considered as the cores of a dynamic game: (i) mode of play; (ii) equilibrium type; and (iii) information pattern. Explanation of these three aspects is provided in the following paragraphs; the diagram in Fig. 1 depicts the relationship between the three aspects.

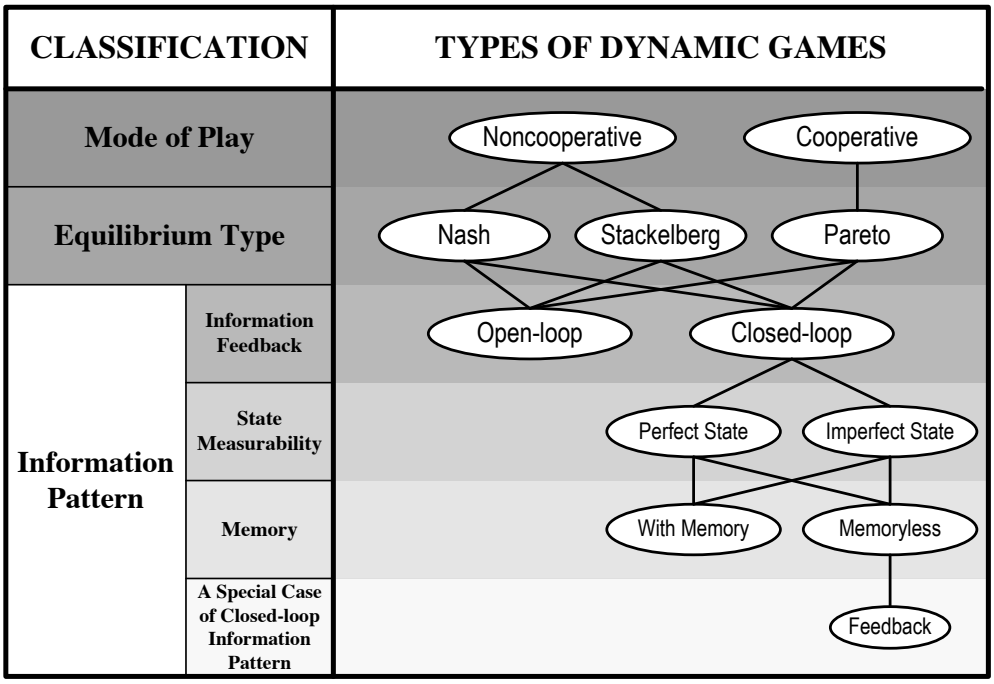

Fig. 1. Classification of dynamic games

The 'mode of play' [13] describes each player's attitude towards his/her own as well as the other players' interests in a game. It can be either noncooperative or cooperative. Shoham and Leyton-Brown [14] explained that in a noncooperative game players consider themselves individuals and concentrate on pursuing their own interest whilst in the cooperative case each player has a sense of collectivity and attempts to enter into a binding agreement of interest.

The ‘equilibrium type’ concerns each player’s strategy adopted for pursuing his/her goal. Here, a ‘strategy’ of a player is a law that tells him/her which action to take at each instant during the game [15]. In other words, it is a mapping from system states or outputs to a player’s control action [16]. An equilibrium denotes a strategy profile consisting of the strategies of all the game players in which no one is willing to change his/her strategy unilaterally [17]. An equilibrium strategy of a player is therefore his/her strategy that constitutes the equilibrium. Nash equilibrium and Stackelberg equilibrium are two typical equilibria that can be frequently observed in noncooperative games. A Nash equilibrium emerges in situations where each player derives his/her strategy by taking the others’ strategies into account, and all the players act simultaneously. A Stackelberg equilibrium, on the other hand, emerges in situations where one player serves as the leader and the others serve as followers. The leader derives his strategy by taking into account all the followers’ optimal responses, whilst all the followers react to the leader’s action by simply using their individual optimal responses. Reference [11] provides an overview of these two equilibria, and [12] gives more technical detail. A concept comparable to Nash and Stackelberg equilibrium, but which emerges in a cooperative game, is called Pareto equilibrium. Since in a cooperative game the goal of each player is identical, the Pareto equilibrium is considered as a global optimality [16]. In view of this, Pareto equilibrium is also called Pareto optimality.

The ‘information pattern’ describes each player’s knowledge of the states of the game system [12]. The players are defined as possessing the ‘open-loop’ information pattern when only the initial states of the game are known to them. On the contrary, if the dynamic states or outputs are available during the game play, players are considered as having the ‘closed-loop’ information pattern. The closed-loop information pattern can be further characterized by the measurability and memory of the states of the game system. In terms of measurability, the closed-loop information pattern is of ‘perfect state’ if the states of the system are completely accessible; whilst it is of ‘imperfect state’ if only the outputs of the system are available. Regarding the memory feature, the closed-loop information pattern is ‘memoryless’ if only the initial and the current states or outputs are remembered by the players; whilst it is ‘with memory’ if all past values of system

states or outputs can be recalled. Under the 'closed-loop memoryless' information pattern (can be either of 'perfect state' or 'imperfect state'), a special case, known as the 'feedback' information pattern can be defined by placing a restriction that the players will forget the initial states upon the start of the game and rely on merely the current states or outputs for deriving their strategies as the game evolves.

A dynamic game whose state evolution is describable by a differential equation is called a differential game [12]. A linear quadratic game is a special case of a differential game in which the system equation is linear, and all players' cost functions are modelled as containing just affine quadratic terms [13].

There are very few published reports of game theory applied in driver-vehicle dynamics. Ma and Peng [18] developed a method to identify the worst-case performance of a car under simultaneous control of a human driver and a vehicle stability controller. The linear quadratic game framework was used and the driver was assumed to be a disturbance who attempts to obstruct the stability controller. The open-loop Nash strategies were calculated to represent the worst-case control of the two controllers. In a separate study, Tamaddoni *et al*. [19] showed that the linear quadratic game framework can be used in the algorithm design of a vehicle stability controller. The vehicle controller was designed to take the driver's steering input into account based on the perfect state feedback Nash equilibrium. Na and Cole [20] developed a mathematical model to represent the noncooperative steering control between driver and AFS, where the open-loop Nash strategies are derived using two alternative game theoretic approaches: noncooperative Model Predictive Control (MPC) and Linear Quadratic (LQ) dynamic optimization. It was found that the two approaches give identical controller gain arrays. In summary, these three published works primarily focus on the use of a Nash equilibrium strategy in investigating driver-vehicle interaction, whilst the potential of applying a Stackelberg equilibrium as well as Pareto optimality appears not to have been explored.

The objectives of the present paper are to: (i) identify and outline a series of paradigms which might be viable in representing a driver's steering interaction with various AFS collision avoidance controllers; and (ii) derive analytical solutions to these paradigms. In the remainder of the paper, these two objectives are addressed respectively in sections II and III. A case study is provided in section IV where simulated time

histories of driver-AFS interaction in different paradigms are exhibited and described. Conclusions and aspects for further investigation are finally presented in section V.

II. DRIVER-AFS INTERACTIVE STEERING CONTROL PARADIGMS

In this section, the steering interaction between driver and AFS-based collision avoidance control is categorized into four possible paradigms, namely decentralized, noncooperative Nash, noncooperative Stackelberg and cooperative Pareto paradigms. Each features a unique communication manner between the driver and AFS, which in turn leads to distinctive formulation of the two controllers' optimization problems and steering control strategies. This section starts with a description of the steering control scheme of the driver and then the AFS collision avoidance system. On this basis, the four paradigms will be established and elucidated one after another.

A. Driver Steering Control Scheme

There exist various mathematical methods of representing human driver steering control behaviour [8], for example the transfer function method, Proportional-Integral-Derivative (PID) control, Model Predictive Control (MPC) and Linear Quadratic Regulator (LQR). Particular attention here is given to the MPC and LQR which serve as the foundation of the distributed MPC and LQ dynamic optimization approaches to be described in section III. A comparative study of MPC and LQR for modelling driver steering control has been performed by Cole *et al*. [21]. It was found that the two methods bear close similarity in terms of the expressions of driver steering control strategies. Furthermore, it was found that the two methods yield identical numerical results when the preview and control horizons are set to be the same. In light of these observations, the driver steering control scheme used in the present work is constructed as shown in Fig. 2, which is capable of illustrating the operating mechanism of both MPC and LQR-based driver models.

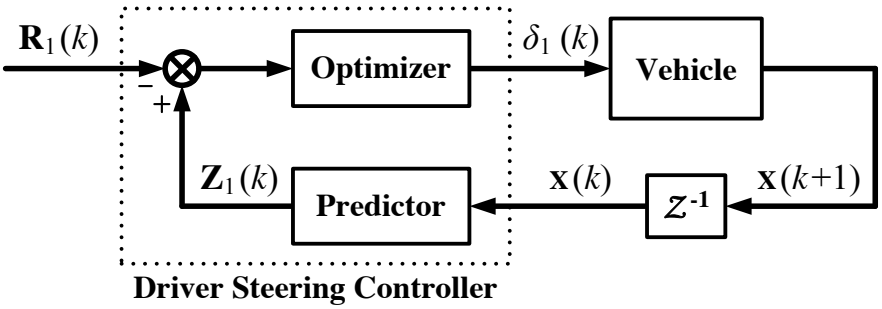

Fig. 2. Driver steering control scheme, applicable to both MPC and LQR methods

When performing a steering control task, the human driver normally previews the road ahead and determines a target path $\mathbf{R}_1(k)$ to follow at each time step k. This $\mathbf{R}_1(k)$ is expressed a sequence of vectors from $\mathbf{r}_1(k)$ to $\mathbf{r}_1(k+N)$ denoting respectively the target vehicle orientations at future time steps up to the driver's preview horizon *N*. Each $\mathbf{r}_1(k+j)$ where $j=0,1,2\ldots N$ consist of three elements: target lateral displacement $r_1^{y}(k+j)$, target lateral displacement integral $r_1^{y_{int}}(k+j)$ and target yaw angle $r_1^{\psi}(k+j)$, that is $\mathbf{r}_1(k+j)=\left\{r_1^{y}(k+j) \quad r_1^{y_{int}}(k+j) \quad r_1^{\psi}(k+j)\right\}$.

In the mean time, the driver perceives the state feedback of the vehicle $\mathbf{x}(k)$ and predicts vehicle future orientation trajectory $\mathbf{Z}_1(k)$ according to his/her knowledge of vehicle dynamics. This process is carried out in the 'Predictor' module shown in Fig. 2. $\mathbf{Z}_1(k)$ here has the same dimension as $\mathbf{R}_1(k)$ but consists of those vehicle orientation elements predicted. In both MPC and LQR methods, derivation of $\mathbf{Z}_1(k)$ is carried out by iterating an equation that represents the driver's knowledge of vehicle dynamics for *N* times [21]. The vehicle in the present research is modelled to operate in the linear regime at constant speed. Consequently, a linear time-invariant 'bicycle' model [20] is used to represent vehicle dynamics, as described in equation (1).

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

In equation (1) state vector $\mathbf{x}(k)$ incorporates vehicle lateral velocity $v(k)$, yaw rate $\omega(k)$, lateral displacement $y(k)$, yaw angle $\psi(k)$ and lateral displacement integral $y_{int}(k)$, that is, $\mathbf{x}(k)=\left\{v(k) \quad \omega(k) \quad y(k) \quad \psi(k) \quad y_{int}(k)\right\}^{\mathrm{T}}$. $\mathbf{A}$ and $\mathbf{B}_1$ are state and input matrices whilst $\mathbf{C}_1$ is the output matrix

that transforms $\mathbf{x}(k)$ into $\mathbf{z}_1(k)$, the vehicle orientation vector that the driver is interested in. The driver is assumed to have full knowledge of such vehicle dynamics. This assumption applies generally to expert drivers and may work for normal drivers who already received extensive training on particular vehicles.

The driver then uses $\mathbf{R}_1(k)$ and $\mathbf{Z}_1(k)$ to determine an optimum steering angle $\delta_1(k)$ in the 'Optimizer' module and finally applies it to the vehicle. Specifically, the driver minimizes a cost function that penalizes both the difference between $\mathbf{Z}_1(k)$ and $\mathbf{R}_1(k)$ and the driver's own steering action. The cost function used in the MPC framework and that in the LQR can be expressed in a unified form as:

$$\begin{aligned} J_1(k) = \sum_{j=0}^{N} [& q_1^{y} \Delta_1^{y}(k+j)^2 + q_1^{y_{int}} \Delta_1^{y_{int}}(k+j)^2 \\ & + q_1^{\psi} \Delta_1^{\psi}(k+j)^2 + p_1 \delta_1(k+j)^2] \end{aligned} \tag{2}$$

where $\Delta_1^{y}(k+j)$, $\Delta_1^{y_{int}}(k+j)$ and $\Delta_1^{\psi}(k+j)$ are respectively the lateral displacement error, lateral displacement error integral and yaw angle error from the driver's view, that is,

$$\Delta_1^{y}(k+j) = z_1^{y}(k+j) - r_1^{y}(k+j) \tag{3a}$$

$$\Delta_1^{y_{int}}(k+j) = z_1^{y_{int}}(k+j) - r_1^{y_{int}}(k+j) \tag{3b}$$

$$\Delta_1^{\psi}(k+j) = z_1^{\psi}(k+j) - r_1^{\psi}(k+j) \tag{3c}$$

q_1^{y}, $q_1^{y_{int}}$ and q_1^{ψ} are corresponding weights. p_1 is the weight on driver steering angle action. Cost function (2) can be further simplified as:

$$J_1(k) = E_1(k, \mathbf{Q}_1) + S_1(k, p_1) \tag{4}$$

where $\mathbf{Q}_1$ is the driver's path-error weighting matrix:

$$\mathbf{Q}_1 = \begin{bmatrix} q_1^{y} & 0 & 0 \\ 0 & q_1^{y_{int}} & 0 \\ 0 & 0 & q_1^{\psi} \end{bmatrix} \tag{5a}$$

$E_1(k, \mathbf{Q}_1)$ is a scalar representing the driver's weighted predicted path errors at time step k:

$$\begin{aligned} E_1(k, \mathbf{Q}_1) = \sum_{j=0}^{N} [& q_1^{y} \Delta_1^{y}(k+j)^2 + q_1^{y_{int}} \Delta_1^{y_{int}}(k+j)^2 \\ & + q_1^{\psi} \Delta_1^{\psi}(k+j)^2] \end{aligned} \tag{5b}$$

and $S_1(k, p_1)$ is a scalar denoting the weighted steering effort:

$$S_1(k, p_1) = \sum_{j=0}^{N} [p_1 \delta_1 (k + j)^2] \tag{5c}$$

Consequently, the optimization problem of the driver can be described as:

$$\begin{aligned} &\min_{\delta_1} \; J_1(k) = E_1(k, \mathbf{Q}_1) + S_1(k, p_1) \\ &s.t. \; \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1 \delta_1(k) \\ \mathbf{z}_1(k) = \mathbf{C}_1 \mathbf{x}(k) \end{cases} \end{aligned} \tag{6}$$

Solving (6) by using either MPC or LQR method results in the driver's optimal steering control strategy as expressed below [21]:

$$\delta_1(k) = \mathbf{K}_1 \left\{ \mathbf{x}(k) \quad \mathbf{R}_1(k) \right\}^{\mathrm{T}} \tag{7}$$

where $\mathbf{K}_1$ is a time-invariant gain array which is a function of system matrices $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{C}_1$, and weights q_1^{y}, $q_1^{y_{int}}$, q_1^{ψ} and p_1.

B. AFS Collision Avoidance Control Scheme

It was noted earlier that the AFS-based collision avoidance control normally involves three steps: object detection, target path planning and evasive steering. As the detection of surrounding objects and planning of target path are not the focus of the present work, and there are already a number of approaches available, for example [22], [23], it is assumed in this article that the desired collision-free path $\mathbf{R}_2(k)$ is already known to the AFS. With regard to the algorithms for evasive steering control, applications of both MPC and LQR exist, for example [24], [25]. In view of this, a structure similar to that of the driver steering controller proposed above is used for representing vehicle AFS collision avoidance system. Therefore the dynamics equation of a linear time-invariant vehicle system under AFS steering control can be expressed as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}_2 \delta_2(k) \\ \mathbf{z}_2(k) &= \mathbf{C}_2 \mathbf{x}(k) \end{aligned} \tag{8}$$

where $\mathbf{A}$, $\mathbf{B}_2$ and $\mathbf{C}_2$ are system matrices, $\delta_2(k)$ the AFS steering input generated at time step k, and $\mathbf{z}_2(k)$ the vehicle orientation vector with which the AFS controller is concerned. Similar to the driver's cost function presented as (2) the AFS controller's cost function can be written as:

$$\begin{aligned} J_2(k) = \sum_{j=0}^{N} [& q_2^{y} \Delta_2^{y}(k+j)^2 + q_2^{y_{int}} \Delta_2^{y_{int}}(k+j)^2 \\ & + q_2^{\psi} \Delta_2^{\psi}(k+j)^2 + p_2 \delta_2(k+j)^2] \end{aligned} \tag{9}$$

where q_2^{y}, $q_2^{y_{int}}$ and q_2^{ψ} are respectively the weights on vehicle lateral deviation Δ_2^{y}, lateral deviation integral $\Delta_2^{y_{int}}$ and yaw deviation Δ_2^{ψ} from the AFS controller's view. p_2 is the weight on AFS steering input δ_2. (9) can be simplified to:

$$J_2(k) = E_2(k, \mathbf{Q}_2) + S_2(k, p_2) \tag{10}$$

where $\mathbf{Q}_2$ is driver's path-following error weighting matrix:

$$\mathbf{Q}_2 = \begin{bmatrix} q_2^{y} & 0 & 0 \\ 0 & q_2^{y_{int}} & 0 \\ 0 & 0 & q_2^{\psi} \end{bmatrix} \tag{11}$$

$E_2(k, \mathbf{Q}_2)$ is a scalar representing the AFS controller's weighted predicted path-following errors at time step k and $S_2(k, p_2)$ is a scalar describing the weighted AFS steering effort. As a result, the optimization problem faced by the AFS controller can be expressed as:

$$\begin{aligned} & \min_{\delta_2} \; J_2(k) = E_2(k, \mathbf{Q}_2) + U_2(k, p_2) \\ & s.t. \; \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_2 \delta_2(k) \\ \mathbf{z}_2(k) = \mathbf{C}_2 \mathbf{x}(k) \end{cases} \end{aligned} \tag{12}$$

Solving the optimization problem (12) yields the AFS controller's optimal steering control strategy:

$$\delta_2(k) = \mathbf{K}_2 \begin{Bmatrix} \mathbf{x}(k) & \mathbf{R}_2(k) \end{Bmatrix}^{\mathrm{T}} \tag{13}$$

where $\mathbf{K}_2$ is a time-invariant gain array which is a function of matrices $\mathbf{A}$, $\mathbf{B}_2$ and $\mathbf{C}_2$, and weights q_2^{y}, $q_2^{y_{int}}$, q_2^{ψ} and p_2.

C. Driver-AFS Decentralized Steering Control Paradigm

In a collision avoidance scenario, the vehicle is subject to the steering control of the driver and the AFS controller simultaneously, and each controller has a target path to follow. Under such circumstances, the most fundamental type of driver-AFS steering interaction, namely the 'decentralized' paradigm, can be established by directly combining the two controllers' steering control schemes together, as shown in Fig. 3.

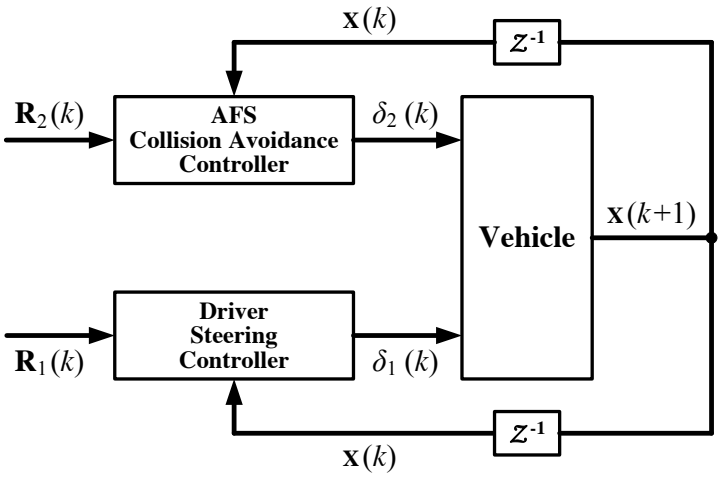

Fig. 3. Driver-AFS decentralized steering control paradigm

It can be seen from Fig. 3 that the state evolution of the vehicle is now governed by:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \tag{14}$$

which indicates that the vehicle state vector $\mathbf{x}$ is influenced by both driver steering angle δ_1 and AFS steering angle δ_2. The AFS collision avoidance controller studied in the decentralized paradigm has the feature that it disregards the driver's steering action whilst solves an optimization problem identical to (12). Such design concept is used in [3], [24] and [25]. In response, the driver may ignore the influence of the AFS steering action as well and views his/her own optimization problem the same as (6). In other words, no communication exists between driver and AFS. As a result, the optimization problems faced by the two controllers can be rewritten as:

$$\text{Driver:} \quad \begin{aligned} &\min_{\delta_1} \ J_1^{\text{Decen}}(k) = E_1(k, \mathbf{Q}_1) + S_1(k, p_1) \\ &s.t. \ \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) \\ \mathbf{z}_1(k) = \mathbf{C}_1\mathbf{x}(k) \end{cases} \end{aligned} \tag{15a}$$

$$\text{AFS:} \quad \begin{aligned} &\min_{\delta_2} \ J_2^{\text{Decen}}(k) = E_2(k, \mathbf{Q}_2) + S_2(k, p_2) \\ &s.t. \ \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}_2(k) = \mathbf{C}_2\mathbf{x}(k) \end{cases} \end{aligned} \tag{15b}$$

Solving (15a) and (15b) either by using the MPC or LQR method gives the decentralized steering control strategies of the two controllers at time step k:

$$\text{Driver:} \quad \delta_1^{\text{Decen}}(k) = \mathbf{K}_1^{Decen} \left\{\mathbf{x}(k) \quad \mathbf{R}_1(k)\right\}^{\text{T}} \tag{16a}$$

$$\text{AFS:} \quad \delta_2^{\text{Decen}}(k) = \mathbf{K}_2^{Decen} \left\{\mathbf{x}(k) \quad \mathbf{R}_2(k)\right\}^{\text{T}} \tag{16b}$$

where $\mathbf{K}_1^{\text{Decen}} = \mathbf{K}_1$ and $\mathbf{K}_2^{\text{Decen}} = \mathbf{K}_2$ hold.

D. Driver-AFS Noncooperative Nash Steering Control Paradigm

In the practice of AFS-based collision avoidance control, there also exists the design that the AFS controller compensates for driver steering action so as to mitigate possible adverse effects due to the driver's erroneous manoeuvres. Such a design concept has been implemented in [2] and [19]. The human driver, on the other hand, may still expect to control the vehicle to follow his/her own target path. Accordingly, the driver may try to estimate the control action applied by AFS and to neutralize its influence by further increasing his/her own handwheel angle input. As a result, communication between driver and AFS in terms of accounting for each others' steering action emerges, and the two controllers' steering strategies converge to the Nash equilibrium defined in noncooperative game theory. Such form of driver-AFS interaction is diagrammatically described in Fig. 4, namely the driver-AFS noncooperative Nash steering control paradigm.

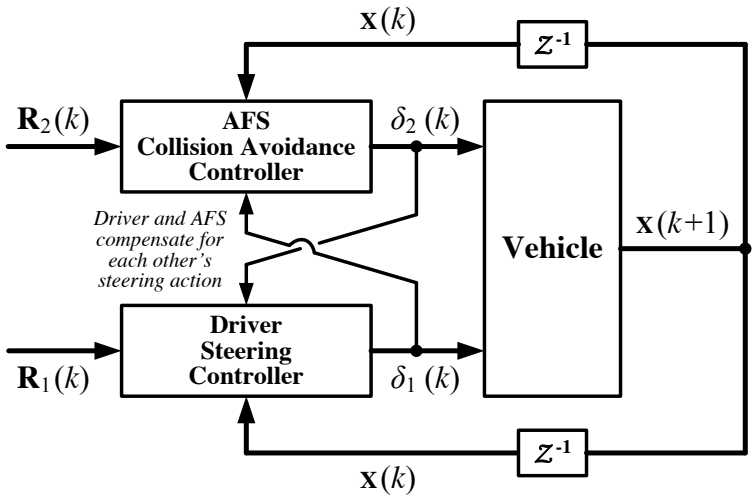

Fig. 4. Driver-AFS noncooperative Nash steering control paradigm

In this case, the dynamics of the driver-AFS system viewed by the driver can be expressed as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}_1(k) &= \mathbf{C}_1\mathbf{x}(k) \end{aligned} \tag{17}$$

whilst that viewed by the AFS controller is:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}_2(k) &= \mathbf{C}_2\mathbf{x}(k) \end{aligned} \tag{18}$$

Accordingly, the optimization problems faced by the two controllers can be respectively written as:

$$\text{Driver:} \quad \min_{\delta_1} J_1^{\text{Nash}}(k) = E_1(k, \mathbf{Q}_1) + S_1(k, p_1) \quad s.t. \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}_1(k) = \mathbf{C}_1\mathbf{x}(k) \end{cases} \tag{19a}$$

$$\text{AFS:} \quad \min_{\delta_2} J_2^{\text{Nash}}(k) = E_2(k, \mathbf{Q}_2) + S_2(k, p_2) \quad s.t. \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}_2(k) = \mathbf{C}_2\mathbf{x}(k) \end{cases} \tag{19b}$$

By comparing (19a) and (19b) with (15a) and (15b), which were formulated according to the decentralized paradigm, it can be seen that the cost functions in the noncooperative Nash and the decentralized cases are identical. This implies that the objectives of controllers do not change. The key difference is that under the noncooperative Nash paradigm the two controllers are aware of the influence of one another's steering action

on the vehicle system. Solving (19a) and (19b) yields the two controllers' respective Nash steering control strategies at time step k:

$$\text{Driver: } \delta_1^{\text{Nash}}(k) = \mathbf{K}_1^{\text{Nash}} \left\{ \mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k) \right\}^{\text{T}} \tag{20a}$$

$$\text{AFS: } \delta_2^{\text{Nash}}(k) = \mathbf{K}_2^{\text{Nash}} \left\{ \mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k) \right\}^{\text{T}} \tag{20b}$$

where both $\mathbf{K}_1^{\text{Nash}}$ and $\mathbf{K}_2^{\text{Nash}}$ are time-invariant gain arrays which are functions of state matrices $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{C}_1$ and $\mathbf{C}_2$, and weights q_1^{y}, $q_1^{y_{int}}$, q_1^{ψ}, q_2^{y}, $q_2^{y_{int}}$, q_2^{ψ}, p_1 and p_2. Here, two remarks on (20a) and (20b) are made. Firstly, it might be questionable that it has been described in Fig. 4 that the driver and the AFS derive their steering strategies by compensating for each other's steering action. However, equations (20a) and (20b) seem not to embody this idea. A brief explanation is that such an idea has been implicitly involved in the procedure for the derivation of the two controllers' steering strategies (20a) and (20b). More details on this will be provided in the next section. Secondly, the reader might be eager to know which particular information pattern was assumed during the derivation of (20a) and (20b). It was suggested in [12] that both the open-loop and the closed-loop information patterns could lead to expressions of strategies as described by (20a) and (20b), however, the elements involved in gain arrays $\mathbf{K}_1^{\text{Nash}}$ and $\mathbf{K}_2^{\text{Nash}}$ differ in these two cases. Investigation in the present work is limited to the open-loop information pattern.

E. Driver-AFS Noncooperative Stackelberg Steering Control Paradigm

When interacting with an AFS controller capable of compensating for driver steering action, as described in Fig. 4, the driver may react in a slightly different manner from that delineated in the noncooperative Nash paradigm. Specifically, the driver may attempt to compensate for the steering control algorithm of the AFS in deriving his/her desirable handwheel angle input, given that the driver knows very well how the AFS collision avoidance control operates. In this event, the interplay between the driver and the AFS controller bears close similarity to that existing in a leader-follower game as explained in section I: the AFS controller plays the role of the follower by making an optimal response to the driver by compensating for the driver's

control action, whilst the driver serves as the leader by taking into consideration the way in which the AFS controller (follower) gives out its optimal response. As a result, steering strategies of the driver (leader) and the AFS (follower) converge to the Stackelberg equilibrium defined in noncooperative game theory. Fig. 5 shows the driver-AFS interaction under such circumstance.

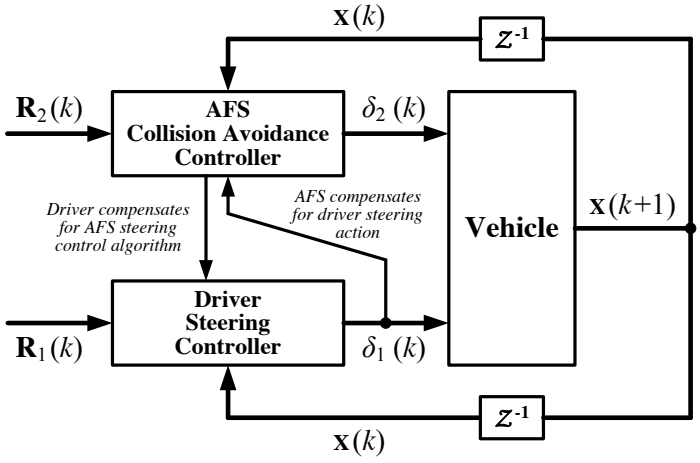

Fig. 5. Driver-AFS noncooperative Stackelberg steering control paradigm

The optimization problems that driver (leader) and AFS (follower) face can be therefore expressed as:

$$\text{Driver (leader):} \quad \min_{\delta_1} J_1^{\text{Stack}}(k) = E_1(k, \mathbf{Q}_1) + S_1(k, p_1) \quad s.t. \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2^{\text{Stack}}(k) \\ \mathbf{z}_1(k) = \mathbf{C}_1\mathbf{x}(k) \end{cases} \tag{21a}$$

$$\text{AFS (follower):} \quad \min_{\delta_2} J_2^{\text{Stack}}(k) = E_2(k, \mathbf{Q}_2) + S_2(k, p_2) \quad s.t. \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}_2(k) = \mathbf{C}_2\mathbf{x}(k) \end{cases} \tag{21b}$$

By comparing (21a) and (21b) with (19a) and (19b), it can be seen that the cost functions used in the Stackelberg case are identical to those in the Nash case. The key difference is that under the Stackelberg paradigm the driver (leader) takes into account the optimal response δ_2^{Stack} of the AFS controller (follower) in minimizing his/her own cost function. Solving (21a) and (21b) by assuming the open-loop information pattern gives the two controllers' open-loop Stackelberg steering control strategies at time step k:

$$\text{Driver (leader):} \quad \delta_1^{\text{Stack}}(k) = \mathbf{K}_1^{\text{Stack}} \{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\text{T}} \tag{22a}$$

$$\text{AFS (follower):}\ \ \delta_2^{\text{Stack}}(k) = \mathbf{K}_2^{\text{Stack}}\left\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\right\}^{\text{T}} \tag{22b}$$

where both $\mathbf{K}_1^{\text{Stack}}$ and $\mathbf{K}_2^{\text{Stack}}$ are time-invariant gain arrays which are functions of system matrices $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{C}_1$ and $\mathbf{C}_2$, and weights q_1^y, $q_1^{y_{int}}$, q_1^{ψ}, q_2^y, $q_2^{y_{int}}$, q_2^{ψ}, p_1 and p_2. At this point, readers might be somewhat confused because it was stated that the follower (AFS controller) uses its optimal response to react to the leader (driver), however, why does the resulting AFS steering control strategy (22b) not seem to show this idea? A short answer is that the derivation of (22b) follows an analytical approach where this idea is implemented as an intermediate step. Detailed explanation will be provided in the next section.

F. Driver-AFS Cooperative Pareto Steering Control Paradigm

In the preceding paragraphs, two noncooperative-game-theory-based steering control paradigms, namely the Nash and Stackelberg paradigms, are delineated. In the present subsection, an alternative driver-AFS interaction paradigm developed based on the principle of Pareto optimality in cooperative game theory is proposed, namely the cooperative Pareto paradigm. Under this paradigm, the AFS collision avoidance controller keeps track of both the driver's steering angle action and the driver's path-following objective whilst the driver is modelled to react to the AFS control in the same manner, as depicted in Fig. 6.

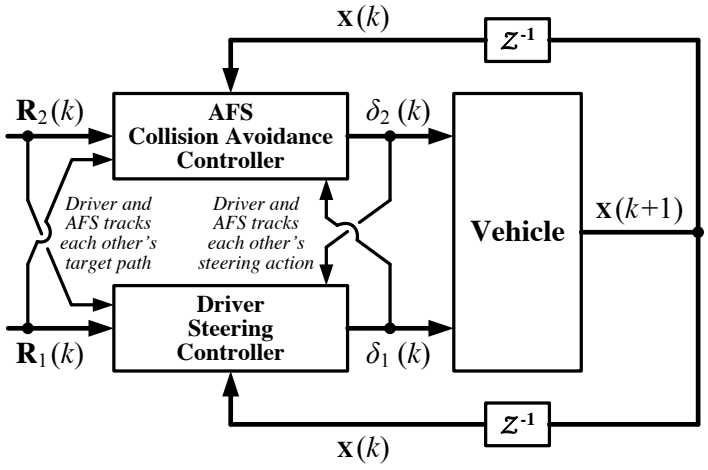

Fig. 6. Driver-AFS cooperative Pareto steering control paradigm

It can be seen in Fig. 6 that two communication channels exist in the Pareto paradigm, such that the driver and the AFS controller get access to one another's target path $\mathbf{R}_1(k)$ or $\mathbf{R}_2(k)$ allows each controller to be able to evaluate the other's path-following error $E_1(k)$ or $E_2(k)$. Accordingly, the two controllers tend to hold a global path-following objective that minimizes $E_1(k)$ and $E_2(k)$ synchronously. The communication of steering angle action in the present paradigm is closely comparable to that exhibited in the noncooperative Nash paradigm. Such communication allows each controller to track how its own as well as the other's steering actions influence the vehicle path-following performance. In light of the explanations made above, the driver and the AFS controllers' optimization problems can be formulated respectively as:

$$\text{Driver:} \quad \begin{aligned} &\min_{\delta_1} \; J_1^{\text{Pareto}}(k) = E_1(k, \mathbf{Q}_1) + E_2(k, \mathbf{Q}_2) + S_1(k, p_1) \\ &s.t. \quad \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}_1(k) = \mathbf{C}_1\mathbf{x}(k) \\ \mathbf{z}_2(k) = \mathbf{C}_2\mathbf{x}(k) \end{cases} \end{aligned} \tag{23a}$$

$$\text{AFS:} \quad \begin{aligned} &\min_{\delta_2} \; J_2^{\text{Pareto}}(k) = E_1(k, \mathbf{Q}_1) + E_2(k, \mathbf{Q}_2) + S_2(k, p_2) \\ &s.t. \quad \begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\delta_1(k) + \mathbf{B}_2\delta_2(k) \\ \mathbf{z}_1(k) = \mathbf{C}_1\mathbf{x}(k) \\ \mathbf{z}_2(k) = \mathbf{C}_2\mathbf{x}(k) \end{cases} \end{aligned} \tag{23b}$$

It can be seen that the driver employs a cost function that minimizes the combination of his own weighted path-following errors E_1 and the AFS controller's E_2 over the driver's steering angle δ_1, and vice versa for the AFS controller. The two controllers also need to know each other's output equations in order to move to the optimization synchronously. Solving (23a) and (23b) by assuming the open-loop information pattern gives the open-loop Pareto steering control strategies at time step k:

$$\text{Driver:} \;\; \delta_1^{\text{Pareto}}(k) = \mathbf{K}_1^{\text{Pareto}} \left\{ \mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k) \right\}^{\text{T}} \tag{24a}$$

$$\text{AFS:} \;\; \delta_2^{\text{Pareto}}(k) = \mathbf{K}_2^{\text{Pareto}} \left\{ \mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k) \right\}^{\text{T}} \tag{24b}$$

where both $\mathbf{K}_1^{\text{Pareto}}$ and $\mathbf{K}_2^{\text{Pareto}}$ are time-invariant gain arrays which are functions of matrices $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{C}_1$ and $\mathbf{C}_2$ and weights q_1^{y}, $q_1^{y_{int}}$, q_1^{ψ}, q_2^{y}, $q_2^{y_{int}}$, q_2^{ψ}, p_1 and p_2. Specifically, the derivation of (24a) and (24b) from optimization problems (23a) and (23b) follows a game theoretic approach similar to that used for

calculating the Nash strategies. More details will be given in the next section. At this point, readers might have a question regarding the feasibility for a driver to detect the AFS controller's target path $\mathbf{R}_2(k)$. A possible way is to generate steering wheel torque feedback that informs the driver of the AFS controller's target.

III. GAME THEORETIC APPROACHES TO DRIVER-AFS INTERACTIVE STEERING CONTROL

In the previous section, four driver-AFS interactive steering paradigms were outlined. It was also revealed that the decentralized steering control strategies are developed using optimal control theory (MPC or LQR) whilst the Nash, Stackelberg and Pareto steering control strategies are derived using game theoretic approaches. In this section, two applicable game theoretic approaches are discussed: the distributed Model Predictive Control (MPC) and the Linear Quadratic (LQ) dynamic optimization approaches.

A. Distributed MPC

The idea of distributed MPC was presented in [26] as a practical approach to industrial process control of large-scale systems that consist of multiple networked subsystems, for example, a chemical plant comprising many reactors. It aims at enabling a compromise between controller performance and computation efficiency. The bridge from distributed MPC to dynamic game theory was examined later by Rawlings and Mayne [27] where its association with Nash equilibrium and Pareto optimality are discussed. The application of distributed MPC to the modelling of noncooperative and cooperative driver-AFS steering interaction is broadly based on the analysis provided in [27], with the procedure for solving controllers' optimization problems following that of Maciejowski [28].

The distributed MPC approach starts with the construction of controllers' cost functions following a two-step procedure: (i) establishment of prediction equations; and (ii) evaluation of path-following errors. It then calculates controllers' steering strategies in two successive steps: (iii) optimization as least-squares

problems; and (iv) derivation of strategies. Treatment of the three game theoretic paradigms involves all these four steps but differ from one another in some detail.

Noncooperative Nash Strategy

Under the noncooperative Nash paradigm, the prediction equation of the driver is established by iterating (17) for N time steps ahead where N denotes the preview horizon. As a result, the driver's prediction equation can be expressed as:

$$\mathbf{Z}_1(k) = \mathbf{\Psi}_1\mathbf{x}(k) + \mathbf{\Theta}_1\mathbf{U}_1(k) + \mathbf{\Omega}_1\mathbf{U}_2(k) \tag{25}$$

where

$$\mathbf{U}_1(k) = \begin{Bmatrix} \delta_1(k) \\ \delta_1(k+1) \\ \vdots \\ \delta_1(k+N-1) \end{Bmatrix}, \ \mathbf{U}_2(k) = \begin{Bmatrix} \delta_2(k) \\ \delta_2(k+1) \\ \vdots \\ \delta_2(k+N-1) \end{Bmatrix},$$

$\mathbf{\Psi}_1$ is a function of matrices $\mathbf{A}$ and $\mathbf{C}_1$, $\mathbf{\Theta}_1$ is a function of $\mathbf{A}$, $\mathbf{B}_1$ and $\mathbf{C}_1$, and $\mathbf{\Omega}_1$ is a function of $\mathbf{A}$, $\mathbf{B}_2$ and $\mathbf{C}_1$. Similarly the AFS collision avoidance controller's prediction equation can be worked out by iterating (18) for N times:

$$\mathbf{Z}_2(k) = \mathbf{\Psi}_2\mathbf{x}(k) + \mathbf{\Theta}_2\mathbf{U}_1(k) + \mathbf{\Omega}_2\mathbf{U}_2(k) \tag{26}$$

The driver's predicted future orientation trajectory $\mathbf{Z}_1(k)$ is then subtracted from the driver's target path $\mathbf{R}_1(k)$ to give the driver's predicted path-error vector. Weighting this vector with $\mathbf{Q}_1$ and computing the square of the Euclidean norm of the weighted vector then yields the driver's weighted path-following errors E_1, as shown in (5b). On this basis, the driver's cost function as described in (19a) can be constructed. The AFS cost function shown in (19b) can be built up in a similar way.

Following Maciejowski [28], optimization problems (19a) and (19b) can be solved as least-squares problems by using QR decomposition which yields (27a) and (27b) below. Details on the intermediate algebraic steps can be found in [20].

$$\mathbf{U}_1(k) = \mathbf{G}_1\{\mathbf{x}(k) \quad \mathbf{R}_1(k)\}^{\mathrm{T}} + \mathbf{L}_1\mathbf{U}_2(k) \tag{27a}$$

$$\mathbf{U}_2(k) = \mathbf{G}_2\{\mathbf{x}(k) \quad \mathbf{R}_2(k)\}^{\mathrm{T}} + \mathbf{L}_2\mathbf{U}_1(k) \tag{27b}$$

where $\mathbf{G}_1$ is a gain array which is a function of $\mathbf{\Psi}_1$, $\mathbf{\Theta}_1$ and driver weights q_1^y, $q_1^{y_{int}}$, q_1^{ψ} and p_1, $\mathbf{L}_1$ is a gain array as a function of $\mathbf{\Theta}_1$, $\mathbf{\Theta}_2$ and driver weights q_1^y, $q_1^{y_{int}}$, q_1^{ψ} and p_1, $\mathbf{G}_2$ is a function of $\mathbf{\Psi}_2$, $\mathbf{\Theta}_2$ and AFS weights q_2^y, $q_2^{y_{int}}$, q_2^{ψ} and p_2, and $\mathbf{L}_2$ is a function of $\mathbf{\Theta}_1$, $\mathbf{\Theta}_2$ and AFS weights q_2^y, $q_2^{y_{int}}$, q_2^{ψ} and p_2. It can be seen that the driver steering input sequence $\mathbf{U}_1$ appearing in (27a) depends on vehicle state $\mathbf{x}$, driver target path $\mathbf{R}_1$, and the AFS controller's steering action sequence $\mathbf{U}_2$, and vice versa for the AFS. Equations (27a) and (27b) embody the communication between driver and AFS in terms of accounting for each other's steering action, as depicted in Fig. 4. They are named the optimal responses of the driver and the AFS controller, respectively. By substituting (27a) and (27b) into one another, the two controllers' open-loop Nash steering sequences can be obtained:

$$\mathbf{U}_1^{\mathrm{Nash}}(k) = \mathbf{\Lambda}_1^{\mathrm{Nash}}\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\mathrm{T}} \tag{28a}$$

$$\mathbf{U}_2^{\mathrm{Nash}}(k) = \mathbf{\Lambda}_2^{\mathrm{Nash}}\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\mathrm{T}} \tag{28b}$$

where both $\mathbf{\Lambda}_1^{\mathrm{Nash}}$ and $\mathbf{\Lambda}_2^{\mathrm{Nash}}$ are functions of $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{L}_1$ and $\mathbf{L}_2$. It can be seen that the driver's Nash steering sequence $\mathbf{U}_1^{\mathrm{Nash}}(k)$ is a vector with a number of N steering angle actions, starting from $\delta_1^{\mathrm{Nash}}(k)$ to $\delta_1^{\mathrm{Nash}}(k+N-1)$. The 'receding horizon' idea [28] is then used which involves taking the first element in $\mathbf{U}_1^{\mathrm{Nash}}(k)$ as the steering angle action to be applied to the vehicle at time step k. Therefore, the Nash strategies described in (20a) and (20b) have the features that $\mathbf{K}_1^{\mathrm{Nash}} = \mathbf{\Lambda}_1^{\mathrm{Nash}}(1, :)$ and $\mathbf{K}_2^{\mathrm{Nash}} = \mathbf{\Lambda}_2^{\mathrm{Nash}}(1, :)$ where the operator $\bullet(1, :)$ indicates extracting the first row of relevant matrix to form a gain array.

Noncooperative Stackelberg Strategy

Under the noncooperative Stackelberg paradigm, the AFS controller (follower) builds up its prediction equation in a similar way to that in the Nash case. Since its cost function is also identical to that in the Nash

case, the AFS controller's optimal response can still be expressed as (27b). In contrast, the driver, as the leader in the Stackelberg paradigm, derives his/her strategy in a more sophisticated manner. The driver first develops an equation identical to (25) following the procedures adopted in the Nash case. The driver then substitutes the AFS optimal response (27b) into (25) to obtain his prediction equation, which gives:

$$\mathbf{Z}_1(k) = \mathbf{\Phi}_1\mathbf{x}(k) + \mathbf{\Gamma}_1\mathbf{U}_1(k) + \mathbf{\Xi}_1\mathbf{R}_2(k) \tag{29}$$

where $\mathbf{\Phi}_1$ is a function of $\mathbf{\Psi}_1$ and $\mathbf{G}_2$, $\mathbf{\Gamma}_1$ is the summation of $\mathbf{\Theta}_1$ and $\mathbf{L}_2$, and $\mathbf{\Xi}_1$ is a function of $\mathbf{\Omega}_1$ and $\mathbf{G}_2$.

Minimizing the driver's cost function (4) with (29) serving as his/her prediction equation yields:

$$\mathbf{U}_1^{\text{Stack}}(k) = \mathbf{\Lambda}_1^{\text{Stack}}\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\text{T}} \tag{30a}$$

Equation (30a) is the open-loop Stackelberg steering sequence of the driver (leader). Since in a Stackelberg game the follower reacts to the leader's action by simply using his/her optimal response, the steering sequence of the AFS controller (follower) can be calculated by substituting (30a) into (27b), which gives:

$$\mathbf{U}_2^{\text{Stack}}(k) = \mathbf{\Lambda}_2^{\text{Stack}}\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\text{T}} \tag{30b}$$

By applying the 'receding horizon' concept to (30a) and (30b), the two controllers' open-loop Stackelberg steering control strategies, as expressed by (22a) and (22b) can be derived where $\mathbf{K}_1^{\text{Stack}} = \mathbf{\Lambda}_1^{\text{Stack}}(1, :)$ and $\mathbf{K}_2^{\text{Stack}} = \mathbf{\Lambda}_2^{\text{Stack}}(1, :)$ hold.

Cooperative Pareto Strategy

Under the cooperative Pareto paradigm, the human driver and the AFS collision avoidance controller enter into a binding agreement in vehicle path-following control. In this event, they share a global prediction equation which is a combination of their individual prediction equations described by (25) and (26):

$$\mathbf{Z}(k) = \mathbf{\Psi}\mathbf{x}(k) + \mathbf{\Theta}\mathbf{U}_1(k) + \mathbf{\Omega}\mathbf{U}_2(k) \tag{31}$$

where

$$\mathbf{Z}(k) = \begin{Bmatrix}\mathbf{Z}_1(k)\\ \mathbf{Z}_2(k)\end{Bmatrix},\ \mathbf{\Psi} = \begin{Bmatrix}\mathbf{\Psi}_1\\ \mathbf{\Psi}_2\end{Bmatrix},\ \mathbf{\Theta} = \begin{Bmatrix}\mathbf{\Theta}_1\\ \mathbf{\Theta}_2\end{Bmatrix} \text{ and } \mathbf{\Omega} = \begin{Bmatrix}\mathbf{\Omega}_1\\ \mathbf{\Omega}_2\end{Bmatrix}.$$

Equation (31) is then subtracted from the combination of the two controllers' target paths $\mathbf{R}(k) = \{\mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\mathrm{T}}$ to give a vector that incorporates both controllers' path errors. Weighting the vector using $\mathbf{Q}_1$ and $\mathbf{Q}_2$ and computing the square of the Euclidean norm of the weighted vector yield the weighted path-following objective $E_1 + E_2$. On this basis, the two controllers' cost functions respectively shown in (23a) and (23b) can be constructed. Following a similar procedure to that used for solving the optimization problems in the Nash case, the driver and AFS controller's optimal responses under the Pareto paradigm can be obtained:

$$\mathbf{U}_1(k) = \mathbf{M}_1\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\mathrm{T}} + \mathbf{N}_1\mathbf{U}_2(k) \tag{32a}$$

$$\mathbf{U}_2(k) = \mathbf{M}_2\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\mathrm{T}} + \mathbf{N}_2\mathbf{U}_2(k) \tag{32b}$$

The two controllers' optimal responses (32a) and (32b) are then substituted into each other to give their open-loop Pareto steering sequences:

$$\mathbf{U}_1^{\mathrm{Pareto}}(k) = \mathbf{\Lambda}_1^{\mathrm{Pareto}}\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\mathrm{T}} \tag{33a}$$

$$\mathbf{U}_2^{\mathrm{Pareto}}(k) = \mathbf{\Lambda}_2^{\mathrm{Pareto}}\{\mathbf{x}(k) \quad \mathbf{R}_1(k) \quad \mathbf{R}_2(k)\}^{\mathrm{T}} \tag{33b}$$

Applying the 'receding horizon' idea finally converts (33a) and (33b) into the open-loop Pareto steering control strategies as shown in (24a) and (24b), where $\mathbf{K}_1^{\mathrm{Pareto}} = \mathbf{\Lambda}_1^{\mathrm{Pareto}}(1, :)$ and $\mathbf{K}_2^{\mathrm{Pareto}} = \mathbf{\Lambda}_2^{\mathrm{Pareto}}(1, :)$ hold.

B. LQ Dynamic Optimization

The LQ dynamic optimization approach is applicable to game theoretic models where the system dynamics are represented using a linear differential equation and the cost functions of players contain just affine quadratic terms [13]. In this subsection, the procedures for adopting this approach to the driver and AFS controller's game theoretic steering control strategies will be discussed. The analysis presented hereinafter is generally based on Basar and Olsder [12], with the handling of the optimization problems following the method described by Lewis *et al*. [29].

The LQ dynamic optimization approach starts with the construction of controllers' cost functions

following a two-step procedure: (i) establishment of the environment equation; and (ii) evaluation of path-following errors. It then calculates the steering strategies of the controllers in two successive steps: (iii) optimization as Lagrange multiplier problems; and (iv) derivation of strategies. In the following paragraphs, these four steps will be applied to the three game theoretic paradigms in turn.

Noncooperative Nash Strategy

Under the noncooperative Nash paradigm, an equation describing the environment in which the driver and AFS act is required. This equation should be able to represent both the evolution of the dynamics of the driver-AFS system and the updating of the two controllers' target paths. Following Sharp and Valtetsiotis [30], the updating of the driver's target path can be modelled using a shift register as follows:

$$\mathbf{R}_1(k+1) = \mathbf{A}_r \mathbf{R}_1(k) + \mathbf{B}_r \mathbf{r}_1^{\text{next}}(k) \tag{34}$$

where

$$\mathbf{R}_1(k) = \begin{Bmatrix} \mathbf{r}_1(k) \\ \mathbf{r}_1(k+1) \\ \vdots \\ \mathbf{r}_1(k+N) \end{Bmatrix}, \ \mathbf{A}_r = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ \mathbf{0} & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \mathbf{B}_r = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \text{ and } \mathbf{r}_1^{next}(k) = \mathbf{r}_1(k+N+1).$$

Here $\mathbf{R}_1(k)$ is the driver's target path previewed at time step k and $\mathbf{r}_1^{next}(k)$ is the driver's demanded vehicle orientation that will become $\mathbf{r}_1(k+N)$ at the next time step. It can be seen that by using (34) at each step, the driver's target path can be updated in time. Similarly, the AFS controller's target path updating can be expressed as:

$$\mathbf{R}_2(k+1) = \mathbf{A}_r \mathbf{R}_2(k) + \mathbf{B}_r \mathbf{r}_2^{\text{next}}(k) \tag{35}$$

Consequently, the environment equation can be built up by merging (34) and (35) into the system dynamics equation (14) which gives:

$$\mathbf{w}(k+1) = \mathbf{A}_w \mathbf{w}(k) + \mathbf{B}_{w1}\delta_1(k) + \mathbf{B}_{w2}\delta_2(k) + \mathbf{F}\mathbf{R}^{\text{next}}(k) \tag{36}$$

where

$$\mathbf{w}(k)=\begin{Bmatrix}\mathbf{x}(k)\\ \mathbf{R}_1(k)\\ \mathbf{R}_2(k)\end{Bmatrix},\ \mathbf{R}^{\text{next}}(k)=\begin{Bmatrix}\mathbf{r}_1^{\text{next}}(k)\\ \mathbf{r}_2^{\text{next}}(k)\end{Bmatrix},\ \mathbf{A}_{\text{w}}=\begin{bmatrix}\mathbf{A} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_{\text{r}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{\text{r}}\end{bmatrix},\ \mathbf{B}_{\text{w1}}=\begin{bmatrix}\mathbf{B}_1\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix},\ \mathbf{B}_{\text{w2}}=\begin{bmatrix}\mathbf{B}_2\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix},\ \text{and}\ \mathbf{F}=\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{bmatrix}.$$

Since the state vector $\mathbf{w}$ in (36) contains both vehicle state variables and the two controller's target paths, the path-following errors of each controller can be predicted by pre-multiplying $\mathbf{w}$ by a specific transformation matrix. For example, if a transformation matrix $\mathbf{H}_1$ is defined as:

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

we then have

$$\mathbf{H}_1\mathbf{w}(k)=\begin{bmatrix}\Delta_1^{y}(k)\\ \Delta_1^{y_{int}}(k)\\ \Delta_1^{\psi}(k)\end{bmatrix}$$

where $\Delta_1^{y}(k)$, $\Delta_1^{y_{int}}(k)$ and $\Delta_1^{\psi}(k)$ are the human driver's lateral displacement error, lateral displacement error integral and yaw angle error as described in (3a), (3b) and (3c), respectively. By further weighting $\mathbf{H}_1\mathbf{w}(k)$ using $\mathbf{Q}_1$, the driver's cost function shown in (19a) can be constructed. The AFS controller's cost function (19b) can be built up in a similar way by defining a transformation matrix $\mathbf{H}_2$.

Following Basar and Olsder [12], the optimization to each controller's path-following performance under the noncooperative Nash paradigm can be formulated as a Lagrange multiplier problem. This involves treating the environment equation (36) as an equality constraint, and substituting it into the original cost function with a Lagrange multiplier associated to give an augmented cost function. By zeroing the partial derivatives of the augmented cost function with respect to steering angle action, system state and the multiplier respectively, the Lagrange multiplier problem can be translated into a two-point boundary-value problem shown as follows:

$$\delta_i^{\text{Nash}}(k+j)=-p_i^{-1}\mathbf{B}_{\text{w}i}^{\text{T}}\boldsymbol{\lambda}_i^{\text{Nash}}(k+j+1) \tag{37a}$$

$$\boldsymbol{\lambda}_i^{\text{Nash}}(k+j)=\mathbf{H}_i^{\text{T}}\mathbf{Q}_i\mathbf{H}_i\mathbf{w}(k+j)+\mathbf{A}_{\text{w}}^{\text{T}}\boldsymbol{\lambda}_i^{\text{Nash}}(k+j+1) \tag{37b}$$

$$\mathbf{w}(k+1) = \mathbf{A}_{\mathrm{w}}\mathbf{w}(k) + \mathbf{B}_{\mathrm{w}1}\delta_1^{\mathrm{Nash}}(k) + \mathbf{B}_{\mathrm{w}2}\delta_2^{\mathrm{Nash}}(k) \tag{37c}$$

with boundary conditions:

$$\mathbf{w}(k) \text{ is given} \tag{37d}$$

$$\boldsymbol{\lambda}_i^{\mathrm{Nash}}(k+N) = \mathbf{H}_i^{\mathrm{T}}\mathbf{Q}_i\mathbf{H}_i\mathbf{w}(k+N) \tag{37e}$$

where $i = 1, 2$, $j = 0, 1, 2 \ldots N$, and $\boldsymbol{\lambda}_i^{\mathrm{Nash}}$ is the Lagrange multiplier associated with a controller's cost function ($\boldsymbol{\lambda}_1^{\mathrm{Nash}}$ for the driver and $\boldsymbol{\lambda}_2^{\mathrm{Nash}}$ for the AFS). Lewis *et al*. [22] suggests that the 'sweep method' can be used to solve the two-point boundary-value problem of the form described above, which involves assuming a linear relation between the environment system state $\mathbf{w}$ and the Lagrange multiplier $\boldsymbol{\lambda}_i^{\mathrm{Nash}}$ for each controller:

$$\boldsymbol{\lambda}_i^{\mathrm{Nash}}(k+j) = \mathbf{T}_i^{\mathrm{Nash}}(k+j)\mathbf{w}(k+j) \tag{37f}$$

Due to space limitation the intermediate algebraic steps of the solution are omitted whilst the resulting Nash steering sequences are given in (38a) and (38b). Details on the derivation can be found in [20]:

$$\begin{aligned}\delta_1^{\mathrm{Nash}}(k+j) = &-p_1^{-1}\mathbf{B}_{\mathrm{w}1}^{\mathbf{T}}\mathbf{T}_1^{\mathrm{Nash}}(k+j+1) \\ &\cdot[\mathbf{I} + \mathbf{B}_{\mathrm{w}1}p_1^{-1}\mathbf{B}_{\mathrm{w}1}^{\mathbf{T}}\mathbf{T}_1^{\mathrm{Nash}}(k+j+1) \\ &+ \mathbf{B}_{\mathrm{w}2}p_2^{-1}\mathbf{B}_{\mathrm{w}2}^{\mathbf{T}}\mathbf{T}_2^{\mathrm{Nash}}(k+j+1)]^{-1}\mathbf{A}_{\mathrm{w}}\mathbf{w}(k+j)\end{aligned} \tag{38a}$$

$$\begin{aligned}\delta_2^{\mathrm{Nash}}(k+j) = &-p_2^{-1}\mathbf{B}_{\mathrm{w}2}^{\mathbf{T}}\mathbf{T}_2^{\mathrm{Nash}}(k+j+1) \\ &\cdot[\mathbf{I} + \mathbf{B}_{\mathrm{w}1}p_1^{-1}\mathbf{B}_{\mathrm{w}1}^{\mathbf{T}}\mathbf{T}_1^{\mathrm{Nash}}(k+j+1) \\ &+ \mathbf{B}_{\mathrm{w}2}p_2^{-1}\mathbf{B}_{\mathrm{w}2}^{\mathbf{T}}\mathbf{T}_2^{\mathrm{Nash}}(k+j+1)]^{-1}\mathbf{A}_{\mathrm{w}}\mathbf{w}(k+j)\end{aligned} \tag{38b}$$

where $\mathbf{T}_1^{\mathrm{Nash}}(k+j)$ and $\mathbf{T}_2^{\mathrm{Nash}}(k+j)$ can be computed using the coupled Riccati equations shown as below:

$$\begin{aligned}\mathbf{T}_1^{\mathrm{Nash}}(k+j) = &\mathbf{H}_1^{\mathrm{T}}\mathbf{Q}_1\mathbf{H}_1 \\ &+ \mathbf{A}_{\mathrm{w}}\mathbf{T}_1^{\mathrm{Nash}}(k+j+1)[\mathbf{I} + \mathbf{B}_{\mathrm{w}1}p_1^{-1}\mathbf{B}_{\mathrm{w}1}^{\mathbf{T}}\mathbf{T}_1^{\mathrm{Nash}}(k+j+1) \\ &+ \mathbf{B}_{\mathrm{w}2}p_2^{-1}\mathbf{B}_{\mathrm{w}2}^{\mathbf{T}}\mathbf{T}_2^{\mathrm{Nash}}(k+j+1)]^{-1}\mathbf{A}_{\mathrm{w}}\end{aligned} \tag{39a}$$

$$\begin{aligned}\mathbf{T}_2^{\mathrm{Nash}}(k+j) = &\mathbf{H}_2^{\mathrm{T}}\mathbf{Q}_2\mathbf{H}_2 \\ &+ \mathbf{A}_{\mathrm{w}}\mathbf{T}_2^{\mathrm{Nash}}(k+j+1)[\mathbf{I} + \mathbf{B}_{\mathrm{w}1}p_1^{-1}\mathbf{B}_{\mathrm{w}1}^{\mathbf{T}}\mathbf{T}_1^{\mathrm{Nash}}(k+j+1) \\ &+ \mathbf{B}_{\mathrm{w}2}p_2^{-1}\mathbf{B}_{\mathrm{w}2}^{\mathbf{T}}\mathbf{T}_2^{\mathrm{Nash}}(k+j+1)]^{-1}\mathbf{A}_{\mathrm{w}}\end{aligned} \tag{39b}$$

By applying the 'receding horizon' idea [28] to (38a) and (38b), the two controllers' open-loop Nash steering control strategies as stated in (20a) and (20b) can be derived where the time-invariant gain arrays $\mathbf{K}_1^{\mathrm{Nash}}$ and $\mathbf{K}_2^{\mathrm{Nash}}$ have the form:

$$\mathbf{K}_1^{\text{Nash}} = -p_1^{-1}\mathbf{B}_{\text{w}1}^{\mathbf{T}}\mathbf{T}_1^{\text{Nash}}(k+1) \cdot [\mathbf{I} + \mathbf{B}_{\text{w}1}p_1^{-1}\mathbf{B}_{\text{w}1}^{\mathbf{T}}\mathbf{T}_1^{\text{Nash}}(k+1) + \mathbf{B}_{\text{w}2}p_2^{-1}\mathbf{B}_{\text{w}2}^{\mathbf{T}}\mathbf{T}_2^{\text{Nash}}(k+1)]^{-1}\mathbf{A}_{\text{w}} \tag{40a}$$

$$\mathbf{K}_2^{\text{Nash}} = -p_2^{-1}\mathbf{B}_{\text{w}2}^{\mathbf{T}}\mathbf{T}_2^{\text{Nash}}(k+1) \cdot [\mathbf{I} + \mathbf{B}_{\text{w}1}p_1^{-1}\mathbf{B}_{\text{w}1}^{\mathbf{T}}\mathbf{T}_1^{\text{Nash}}(k+1) + \mathbf{B}_{\text{w}2}p_2^{-1}\mathbf{B}_{\text{w}2}^{\mathbf{T}}\mathbf{T}_2^{\text{Nash}}(k+1)]^{-1}\mathbf{A}_{\text{w}} \tag{40b}$$

Noncooperative Stackelberg Strategy

Under the noncooperative Stackelberg paradigm, the two controllers construct their cost functions in the same way as in the Nash case. Moreover, since the follower (AFS controller) reacts to the leader (driver) by simply applying its optimal response, the Lagrange multiplier problem of the AFS can be formulated the same as in the Nash case. As a result, equations (37a) and (37b) still hold for the AFS controller ($i = 2$). However, since the driver takes into account the optimal response of the AFS controller, the driver's Lagrange multiplier problem differs from that defined under the Nash paradigm. Specifically, the driver uses the environment equation (36), and expressions (37a) and (37b) at $i = 2$ as equality constraints to his cost function. This in turn yields a more complicated two-point boundary-value problem. In this case a simple equation as presented in (37f) can no longer facilitate the solution of the problem [31]. Using the approach of Hungerländer and Neck [32], more sophisticated hypothetic equations are used in the sweep method for deriving the open-loop Stackelberg strategies. It was demonstrated by Na and Cole [33] that the two controllers' Stackelberg strategies derived using the LQ dynamic optimization approach have the identical analytical expressions to those using the distributed MPC approach.

Cooperative Pareto Strategy

Under the cooperative Pareto paradigm, the two controllers share a common objective as a linear combination of their individual path-following objectives. Hence, each controller penalizes the summation of $\mathbf{H}_1\mathbf{w}(k)$ and $\mathbf{H}_2\mathbf{w}(k)$. Since the two controllers communicate in terms of getting access to one another's control action, the Lagrange-multiplier-based optimization can be applied in a similar way to that in the Nash

case. In view of this, the environment equation (36) is still used as a constraint for constructing each controller's augmented cost function. By rendering relevant partial derivatives of the two controllers' augmented cost functions to vanish, a two-point boundary-value problem similar to that defined in the Nash paradigm is yielded:

$$\delta_i^{\text{Pareto}}(k+j) = -p_i^{-1}\mathbf{B}_{\text{w}i}^{\mathbf{T}}\boldsymbol{\lambda}_i^{\text{Pareto}}(k+j+1) \tag{41a}$$

$$\begin{aligned}\boldsymbol{\lambda}_i^{\text{Pareto}}(k+j) = &[\mathbf{H}_1^{\text{T}}\mathbf{Q}_1\mathbf{H}_1 + \mathbf{H}_2^{\text{T}}\mathbf{Q}_2\mathbf{H}_2] \\ &\cdot \mathbf{w}(k+j) + \mathbf{A}_{\text{w}}^{\mathbf{T}}\boldsymbol{\lambda}_i^{\text{Pareto}}(k+j+1)\end{aligned} \tag{41b}$$

$$\mathbf{w}(k+1) = \mathbf{A}_{\text{w}}\mathbf{w}(k) + \mathbf{B}_{\text{w}1}\delta_1^{\text{Pareto}}(k) + \mathbf{B}_{\text{w}2}\delta_2^{\text{Pareto}}(k) \tag{41c}$$

with boundary conditions:

$$\mathbf{w}(k) \text{ is given} \tag{41d}$$

$$\boldsymbol{\lambda}_i^{\text{Pareto}}(k+N) = [\mathbf{H}_1^{\text{T}}\mathbf{Q}_1\mathbf{H}_1 + \mathbf{H}_2^{\text{T}}\mathbf{Q}_2\mathbf{H}_2]\mathbf{w}(k+N) \tag{41e}$$

where $i = 1, 2$, $j = 0, 1, 2 \ldots N$, and $\boldsymbol{\lambda}_i^{\text{Pareto}}$ is the Lagrange multiplier ($\boldsymbol{\lambda}_1^{\text{Pareto}}$ for the driver and $\boldsymbol{\lambda}_2^{\text{Pareto}}$ for the AFS). On this basis, the 'sweep method' is used and a linear relation between the environment system state $\mathbf{w}$ and the Lagrange multiplier $\boldsymbol{\lambda}_i^{\text{Pareto}}$ is assumed for each controller:

$$\boldsymbol{\lambda}_i^{\text{Pareto}}(k+j) = \mathbf{T}_i^{\text{Pareto}}(k+j)\mathbf{w}(k+j) \tag{41f}$$

Solving the two-point boundary-value problem defined from (41a) to (41f) gives the two controllers' Pareto steering sequences:

$$\begin{aligned}\delta_1^{\text{Pareto}}(k+j) = &-p_1^{-1}\mathbf{B}_{\text{w}1}^{\mathbf{T}}\mathbf{T}_1^{\text{Pareto}}(k+j+1) \\ &\cdot[\mathbf{I} + \mathbf{B}_{\text{w}1}p_1^{-1}\mathbf{B}_{\text{w}1}^{\mathbf{T}}\mathbf{T}_1^{\text{Pareto}}(k+j+1) \\ &+ \mathbf{B}_{\text{w}2}p_2^{-1}\mathbf{B}_{\text{w}2}^{\mathbf{T}}\mathbf{T}_2^{\text{Pareto}}(k+j+1)]^{-1}\mathbf{A}_{\text{w}}\mathbf{w}(k+j)\end{aligned} \tag{42a}$$

$$\begin{aligned}\delta_2^{\text{Pareto}}(k+j) = &-p_2^{-1}\mathbf{B}_{\text{w}2}^{\mathbf{T}}\mathbf{T}_2^{\text{Pareto}}(k+j+1) \\ &\cdot[\mathbf{I} + \mathbf{B}_{\text{w}1}p_1^{-1}\mathbf{B}_{\text{w}1}^{\mathbf{T}}\mathbf{T}_1^{\text{Pareto}}(k+j+1) \\ &+ \mathbf{B}_{\text{w}2}p_2^{-1}\mathbf{B}_{\text{w}2}^{\mathbf{T}}\mathbf{T}_2^{\text{Pareto}}(k+j+1)]^{-1}\mathbf{A}_{\text{w}}\mathbf{w}(k+j)\end{aligned} \tag{42b}$$

where $\mathbf{T}_1^{\text{Pareto}}(k+j)$ and $\mathbf{T}_2^{\text{Pareto}}(k+j)$ are computed using relevant coupled Riccati equations.

Applying the 'receding horizon' idea finally converts (42a) and (42b) into the open-loop Pareto steering control strategies as shown in (24a) and (24b), where

$$\mathbf{K}_1^{\text{Pareto}} = -p_1^{-1}\mathbf{B}_{\text{w1}}^{\mathbf{T}}\mathbf{T}_1^{\text{Pareto}}(k+1) \cdot[\mathbf{I}+\mathbf{B}_{\text{w1}}p_1^{-1}\mathbf{B}_{\text{w1}}^{\mathbf{T}}\mathbf{T}_1^{\text{Pareto}}(k+1)+\mathbf{B}_{\text{w2}}p_2^{-1}\mathbf{B}_{\text{w2}}^{\mathbf{T}}\mathbf{T}_2^{\text{Pareto}}(k+1)]^{-1}\mathbf{A}_{\text{w}} \tag{43a}$$

$$\mathbf{K}_2^{\text{Pareto}} = -p_2^{-1}\mathbf{B}_{\text{w2}}^{\mathbf{T}}\mathbf{T}_2^{\text{Pareto}}(k+1) \cdot[\mathbf{I}+\mathbf{B}_{\text{w1}}p_1^{-1}\mathbf{B}_{\text{w1}}^{\mathbf{T}}\mathbf{T}_1^{\text{Pareto}}(k+1)+\mathbf{B}_{\text{w2}}p_2^{-1}\mathbf{B}_{\text{w2}}^{\mathbf{T}}\mathbf{T}_2^{\text{Pareto}}(k+1)]^{-1}\mathbf{A}_{\text{w}} \tag{43b}$$

IV. CASE STUDY

In this subsection, the driver-AFS steering interaction modelled using the four paradigms described above is studied through simulation. A scenario describing a driver steering counter to the AFS collision avoidance control is designed. Specifically, it is assumed that when an obstacle suddenly appears in front, e.g. a pedestrian running into the road, the vehicle AFS controller initiates a lane change towards the left whilst the driver decides to carry out evasive steering to the right. Similar driver counter-steering behaviours in response to vehicle active steering intervention were observed by Katzourakis *et al*. [7] in a driving simulator experiment. Particular interest in the present study is given to the influence of driver path-error weights, that is q_1^{y}, $q_1^{y_{int}}$ and q_1^{ψ} on his/her steering behaviours and resulting vehicle lateral response. All the other parameters are fixed throughout the study, as described in Table I. In this table the vehicle parameters are measured based on an Opel Signum passenger car [34] whilst the AFS controller path-error weights are determined to maintain decent path-tracking performance whilst allow human drivers to override the AFS control [35].

TABLE I
PARAMETER VALUES FOR SIMULATION STUDY

Symbol	Quantity	Value
m	vehicle mass	1840 kg
I	vehicle yaw moment of inertia	3000 kgm^2
l_a / l_b	distance from vehicle centre of mass to front axle / rear axle	1.136 / 1.663 m
C_f / C_r	cornering stiffness of front axle / rear axle	116000 / 187000 N/rad
G	vehicle overall steering ratio	15.8
U	vehicle longitudinal velocity	20 m/s
q_2^y	AFS controller lateral displacement error weight	6e-2
$q_2^{y_{int}}$	AFS controller lateral displacement error integral weight	0
q_2^{ψ}	AFS controller yaw angle error weight	0
p_1 / p_2	driver / AFS controller steering angle input weight	1
N	preview horizon	2.0 s
T_s	simulation time step	0.01 s

Fig. 7 shows the simulation outcomes from the four paradigms. For each paradigm, the subfigure presented on the left displays the driver and AFS target paths and the simulated vehicle lateral displacement in relation to its longitudinal position, whilst the subfigure on the right depicts the two controllers' steering angles. It can be seen that the driver and the AFS are set to have lane change target paths of the same geometry but in opposite directions, as described respectively using circle and triangle markers in the left subfigure. Within each paradigm four typical sets of driver path-error weight combination $(q_1^y, q_1^{y_{int}}, q_1^{\psi})$ are investigated: (i) the $(q_1^y = 0, q_1^{y_{int}} = 0, q_1^{\psi} = 0)$ set representing that the driver does not care about path-following performance – simulation results in terms of vehicle lateral displacement (left subfigure) and controllers' steering angles (right subfigure) are described using solid lines; (ii) the $(q_1^y = 6\text{e}-2, q_1^{y_{int}} = 0, q_1^{\psi} = 0)$ set indicating that the driver holds identical path-error weight to the AFS controller – described by dashed lines; (iii) the $(q_1^y = 3\text{e}-1, q_1^{y_{int}} = 0, q_1^{\psi} = 0)$ set indicating the driver uses larger lateral displacement error weight – described by dotted lines; and (iv) the $(q_1^y = 6\text{e}-2, q_1^{y_{int}} = 6\text{e}-5, q_1^{\psi} = 0)$ set denoting the driver further penalizes vehicle lateral displacement error integral – described by dash-dot lines.

Under the decentralized paradigm it can be observed that when the driver applies zero path-error weights (driver weight set (i), see solid lines in the left and right subfigures), no driver steering action is generated and

the vehicle is controlled by the AFS to track its triangle-marked target path. When the driver uses a weight set identical to AFS controller's (driver weight set (ii), see dashed lines), the two controllers give opposite steering angles, and the vehicle travels along the central line. When the driver increases his lateral displacement error weight to $q_1^y = 3\text{e-}1$ (driver weight set (iii), described by dotted lines), both controllers' steering angles increase and the vehicle gets closer to the driver's target. However, large vehicle path-following error can be witnessed. As the driver further introduces integral control of vehicle lateral displacement using $q_1^{y_{int}} = 6\text{e-}5$ (driver weight set (iv), described by dash-dot lines), the vehicle converges to the driver's target path at the expense of larger driver steering angles.

The influence of driver path-error weights in the noncooperative Nash paradigm is comparable to that in the decentralized event: larger driver lateral displacement error weight q_1^y causes the vehicle to travel closer to the driver's target path at an expense of larger steering angles (see dotted lines in comparison to dashed lines). The use of lateral displacement error integral weight $q_1^{y_{int}} = 6\text{e-}5$ leads to more accurate path-following with less steady-state errors (see dash-dot lines in the left and right subfigures). Nevertheless, it can be seen clearly that under each driver weight sets examined, both the driver and the AFS steering angles are larger than those in the decentralized case. This is due to the communication in steering actions as depicted in Fig. 4: Featuring compensation for the driver's steering control, the AFS controller tends to add an additional amount of steering angle action to that determined following the decentralized strategy. In response, the driver also increases his/her steering angle input for neutralizing the AFS steering control. As a result, the two controllers' steering actions grow simultaneously until the Nash equilibrium is achieved.

In the noncooperative Stackelberg paradigm, it can be found that when the driver's path-error weight is identical to the AFS controllers' (driver weight set (ii), see dashed lines), the trajectory of the vehicle is closer to the AFS target path. This is different from those displayed in the decentralized and Nash cases. Such a phenomenon embodies the asymmetry in the roles that the driver and the AFS play in the Stackelberg paradigm, which in turn implies that as the leader the driver needs to use larger path-error weights than the

AFS so as to keep the vehicle travelling along the central line. By the same token, the vehicle path resulting from $q_1^y = 3e-1$ (driver weight set (iii), see dotted lines) is also closer to the AFS target path in comparison to the Nash case. Penalization of vehicle lateral displacement error integral (driver weight set (iv), see dash-dot lines) allows the vehicle to track the driver's target path firmly.

In the cooperative Pareto paradigm, each controller penalizes both its own and the other one's path errors. As a result, when the driver is set to use zero weights (driver weight set (i), described by solid lines), both the driver and the AFS turn to focus on minimizing the AFS controller's path errors. The two controllers generate identical steering angles to control the vehicle following the AFS target path. When the driver uses the same weight as the AFS (driver weight set (ii), described by dashed lines), the vehicle travels along the central line which correspond to zero steering actions. An increase of the driver's lateral displacement error to $q_1^y = 3e-1$ (driver weight set (iii), see dotted lines) enables the vehicle to get closer to the driver's target path, and the introduction of $q_1^{y_{int}} = 6e-5$ (driver weight set (iv), see dash-dot lines) enables the vehicle to travel along the driver's target path.

In summary, each of the four sets of driver controller path-error weights results in distinctive driver steering behaviours and vehicle path-following performance among the four paradigms developed in the present work. In each paradigm, variation of driver path-error weight exerts noticeable influence on the interaction between the driver and vehicle AFS controller.

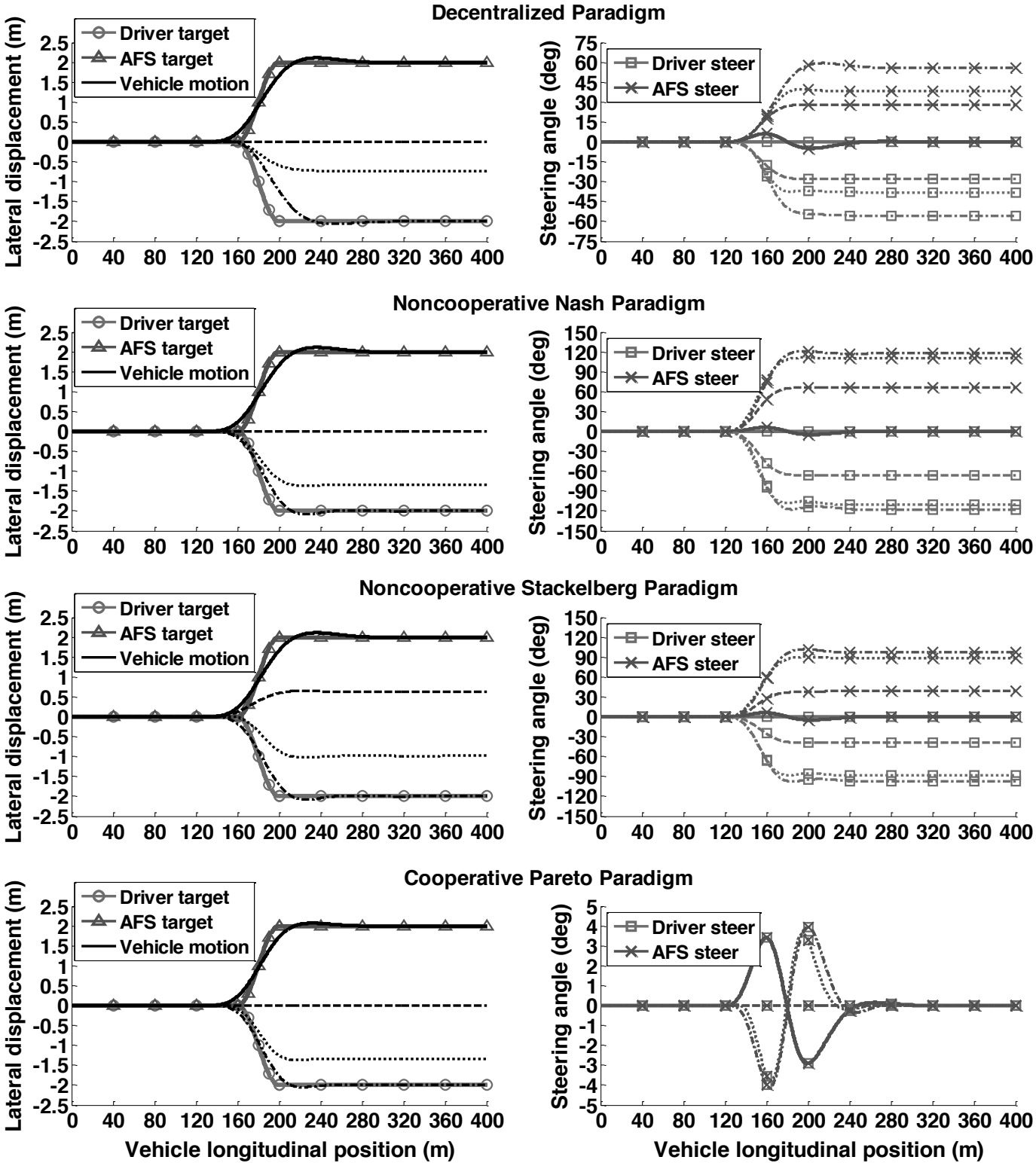

Fig. 7. Simulation results from decentralized, noncooperative Nash, noncooperative Stackelberg and cooperative Pareto paradigms under different driver path-error weight sets. Solid: $(q_1^{y}=0, q_1^{y_{int}}=0, q_1^{\psi}=0)$; dashed: $(q_1^{y}=6\mathrm{e}-2, q_1^{y_{int}}=0, q_1^{\psi}=0)$; dotted: $(q_1^{y}=3\mathrm{e}-1, q_1^{y_{int}}=0, q_1^{\psi}=0)$; dash-dot: $(q_1^{y}=6\mathrm{e}-2, q_1^{y_{int}}=6\mathrm{e}-5, q_1^{\psi}=0)$.

V. CONCLUSION

The first objective of the work described in this paper was to identify and outline a series of paradigms which might be viable in modelling a driver's steering interaction with AFS collision avoidance controllers. To this end, four paradigms were proposed, known as decentralized, noncooperative Nash, noncooperative Stackelberg and cooperative Pareto paradigms, and the following conclusions were reached:

1) The decentralized paradigm represents a driver's steering control in response to an AFS controller that disregards driver steering action. The driver is modelled to ignore the AFS steering action as well.
2) The noncooperative Nash and Stackelberg paradigms represent a driver's steering interaction with an AFS controller that actively compensates for driver steering action. The drivers in these two paradigms are modelled respectively to compensate for the AFS control action and control algorithm.
3) The cooperative Pareto paradigm represents a driver's steering interaction with an AFS controller that further takes into account driver target path. The driver is modelled to react to the AFS in the same way.
4) In the decentralized and the noncooperative cases, each controller intends to minimize its individual path errors, whilst in the cooperative case the two controllers share a global path-following control objective.
5) Under the decentralized paradigm each controller's strategy is independent of the other's target path. However, under the three game theoretic paradigms the two controllers require each other's target paths in formulating their own strategies.

The second objective was to derive the analytical solutions to the four paradigms. Two game theoretic approaches, known as distributed MPC and LQ dynamic optimization were described, with the following conclusions obtained:

6) The two approaches bear some similarity: both construct controllers' cost functions by penalizing path-following errors as well as steering control inputs, and both adopt the 'receding horizon' idea to determine controllers' game theoretic steering control strategies.
7) There are two significant differences: first, distributed MPC requires prediction equations for the

optimization whilst LQ dynamic optimization needs an environment equation; second, distributed MPC formulates the optimization as a least-squares problem whilst LQ dynamic optimization approach treats the optimization as a Lagrange multiplier problem.

8) For each approach, solutions to the three driver-AFS game theoretic paradigms also differ from one another due to the dissimilarity in equilibrium properties.

A case study was performed to illustrate how a driver interacts with a vehicle AFS collision avoidance controller in the four paradigms proposed. The results implied that within a particular paradigm, a variety of driver steering control behaviours can be yielded by varying driver path-error weights. On the other hand, the same driver weight set gives distinct driver behaviours in different paradigms.

In further work, the AFS collision avoidance controller discussed in each of the four paradigms will be implemented in a driving simulator for measuring its exclusive interaction with human drivers. The analytical solution of the driver steering strategy derived in each paradigm will then be used to fit corresponding measured driver steering behaviour. Resultant model fitting errors in both time and frequency domain will be analyzed to demonstrate the validity of the proposed driver-AFS steering control paradigms.

REFERENCES

[1] W. Klier, R. Grossheim, and W. Schuster, "Control algorithms for superposition steering systems to adapt and improve steering characteristics," Society of Automobile Engineers, Warrendale, PA, USA, SAE Tech. Paper 2005-01-1266, 2005.

[2] S. J. Anderson, S. C. Peters, K. D. Iagnemma, and T. E. Pilutti, "A unified approach to semi-autonomous control of passenger vehicles in hazard avoidance scenarios," in *Proc. 2009 IEEE Int. Conf. Syst., Man, Cybern.*, San Antonio, U.S.A., 2009

[3] R. Isermann, R. Mannale, and K. Schmitt, "Collision-avoidance systems PRORETA: situation analysis and intervention control," *J. Control. Eng. Practice*, vol. 20, no. 11, pp. 1236-1246, Nov. 2012.

[4] A. Gray, M. Ali, Y. Gao, J. K. Hedrick, and F. Borrelli, "Semi-autonomous vehicle control for road departure and obstacle avoidance," in *Proc. 13th IFAC Symp. Control in Transportation Syst.*, Sofia, Bulgaria, 2013.

[5] E. Bender, K. Landau, and R. Bruder, "Driver reactions in response to automatic obstacle avoiding manoeuvres," in *Proc. 16th World Congr. Ergonom.*, Maastricht, Netherlands, 2006.

[6] M. Itoh, T. Horikome, and T. Inagaki, "Effectiveness and driver acceptance of a semi-autonomous forward obstacle collision avoidance system," in *Proc. Human Factors and Ergonom. Soc. Annu. Meeting*, San Francisco, U.S.A., 2010.

[7] D. Katzourakis, M. Alirezaei, J. C. F. de Winter, M. Corno, R. Happee, A. Ghaffari, and R. Kazemi, "Shared Control for Road Departure Prevention," in *Proc. 2011 IEEE Int. Conf. Syst., Man, Cybern.*, Anchorage, U.S.A., 2011.

[8] M. Mulder, D. A. Abbink and E. R. Boer, "Sharing control with haptics: seamless driver support from manual to automatic control", *Human Factors*, vol. 54, no. 5, pp. 786-798, Oct., 2012.

[9] D. J. Cole, "A path-following driver-vehicle model with neuromuscular dynamics, including measured and simulated responses to a step in steering angle overlay," *Veh. Syst. Dyn.*, vol. 50, no. 4, pp. 573-596, April 2012.

[10] X. Na and D. J. Cole, "Modelling and identification of a driver controlling a vehicle equipped with active steering, where the driver and vehicle have different target paths," in *Proc. 11th Int. Symp. Advanced Veh. Control*, Seoul, Korea, 2012.

[11] J. B. Cruz, "Survey of Nash and Stackelberg equilibrium strategies in dynamic games," *Ann. Economic and Social Measurement*, vol. 4, no. 2, pp. 339-344, April 1975.

[12] T. Başar and G.J. Olsder. *Dynamic Noncooperative Game Theory* (2nd Edition). New York: Academic Press, 1995.

[13] J. Engwerda, *LQ Dynamic Optimization and Differential Games*. Sussex: John Wiley & Sons, 2005, ch 1.

[14] Y. Shoham, and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge: Cambridge University Press, 2009, ch. 3.

[15] E. Rasmusen, *Games and Information: An Introduction to Game Theory (4th Edition)*. London: Wiley-Blackwell, 2006, ch. 1.

[16] E. J. Dockner and R. Neck, "Time consistency, subgame perfectness, solution concepts and information patterns in dynamic models of stabilization policies," in *Advances in Computational Economis*, vol. 22, *Quantitative Economic Policy*, Berlin: Springer-Verlag, 2008, pp. 51-101.

[17] I. K. Geçkil and P. L. Anderson. *Applied Game Theory and Strategic Behavior*. Boca Raton: CRC Press, 2010, ch. 2.

[18] W. Ma and H. Peng, "Worst-case vehicle evaluation methodology — examples on truck rollover/jackknifing and active yaw control systems," *Veh. Syst. Dyn.*, vol. 32, no. 4-5, pp. 389-408, May 1999.

[19] 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.

[20] X. Na and D. J. Cole, "Linear quadratic game and non-cooperative predictive methods for potential application to modelling driver-AFS interactive steering control," *Veh. Syst. Dyn.*, vol. 51, no. 2, pp. 165-198, Feb. 2012.

[21] 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.

[22] C. F. Wu, C. J. Lin, and C. Y. Lee, "Applying a functional neurofuzzy network to real-time lane detection and front-vehicle distance measurement," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 577-589, July 2012.

[23] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, "Constraint-based planning and control for safe, semi-autonomous operation of vehicles," in *Proc. 2012 IEEE Intell. Veh. Symp.*, Alcalá de henares, Spain, 2012.

[24] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans*. *Control Syst*. *Technol*., vol. 15, no. 3, pp. 566-580, May 2007.

[25] D. Soudbakhsh and A. Eskandarian, "A collision avoidance steering controller using Linear Quadratic Regulator," Society of Automobile Engineers, Warrendale, PA, USA, SAE Paper 2010-01-0459, 2010.

[26] A. N. Venkat, "Distributed Model Predictive Control: Theory and Applications," Ph.D. dissertation, Dept. Chemical Eng., Univ. Wisconsin-Madison, Madison, USA, 2006.

[27] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. New York: Nob Hill, 2009, ch. 6.

[28] J. M. Maciejowski, *Predictive Control: With Constraints*. London: Prentice-Hall, 2002, ch. 3.

[29] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control (3rd Edition)*. Hoboken: John Wiley & Sons, 2012, ch. 2.

[30] R. S. Sharp and V. Valtetsiotis, "Optimal preview car steering control," *Veh*. *Syst*. *Dyn*., vol. 35, suppl., pp. 101-117, Dec. 2001.

[31] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, *Matrix Riccati Equations in Control and Systems Theory*. Basel: Birkhäuser, 2003, ch. 6.

[32] P. Hungerländer and R. Neck, "An algorithmic equilibrium solution for n-person dynamic Stackelberg difference games with open-loop information pattern," in *Dynamic modeling and Econometric in Economics and Finance*, vol. 13, *Computational Methods in Economic Dyn*., Berlin: Springer-Verlag, 2011, pp. 197-214.

[33] 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/C-MECH/TR. 101, 2013.

[34] S. Keen, "Modeling Driver Steering Behavior using Multiple-Model Predictive Control," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge UK, 2008.

[35] X. Na, and D. J. Cole, "Investigation of a driver's interaction with an active steering collision avoidance system," in *Proc. 23rd Int. Symp. Dynamics of Vehicles on Roads and Tracks*, Qingdao, China, 2013.

Xiaoxiang Na received the B.Sc. and M.Sc. degrees in automotive engineering from the College of Automotive Engineering, Jilin University, Changchun, China in 2007 and 2009.

He is currently pursuing a Ph.D. degree in driver-vehicle dynamics in the Department of Engineering, University of Cambridge, Cambridge, U.K. His research interests include game theoretic modelling of driver-vehicle interaction and the numeric approaches to dynamic game problems.

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), Cambridge, U.K. in 1985 and 1990.

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