



Modelling the influence of sensory dynamics on linear and nonlinear driver steering control

C. J. Nash & D. J. Cole

To cite this article: C. J. Nash & D. J. Cole (2017): Modelling the influence of sensory dynamics on linear and nonlinear driver steering control, Vehicle System Dynamics, DOI: [10.1080/00423114.2017.1326615](https://doi.org/10.1080/00423114.2017.1326615)

To link to this article: <http://dx.doi.org/10.1080/00423114.2017.1326615>



© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



[View supplementary material](#)



Published online: 19 May 2017.



[Submit your article to this journal](#)



Article views: 167



[View related articles](#)



[View Crossmark data](#)

Modelling the influence of sensory dynamics on linear and nonlinear driver steering control

C. J. Nash  and D. J. Cole 

Department of Engineering, University of Cambridge, Cambridge, UK

ABSTRACT

A recent review of the literature has indicated that sensory dynamics play an important role in the driver–vehicle steering task, motivating the design of a new driver model incorporating human sensory systems. This paper presents a full derivation of the linear driver model developed in previous work, and extends the model to control a vehicle with nonlinear tyres. Various nonlinear controllers and state estimators are compared with different approximations of the true system dynamics. The model simulation time is found to increase significantly with the complexity of the controller and state estimator. In general the more complex controllers perform best, although with certain vehicle and tyre models linearised controllers perform as well as a full nonlinear optimisation. Various extended Kalman filters give similar results, although the driver's sensory dynamics reduce control performance compared with full state feedback. The new model could be used to design vehicle systems which interact more naturally and safely with a human driver.

ARTICLE HISTORY

Received 5 January 2017
Accepted 28 April 2017

KEYWORDS


Driver behaviour; state estimation; nonlinear control; biodynamics; optimal control; driver–vehicle systems

1. Introduction

Automotive engineers are increasingly developing semi-autonomous systems which interact with the driver to improve comfort, safety and driveability. However, current understanding of the role of the human driver in the vehicle control task is limited, so it is difficult to predict how drivers will respond to vehicle systems without extensive testing. Previous research has focussed on developing empirical driver models for specific circumstances, which are generally limited in their scope and predictive ability [1]. More recently, optimal control theory has been used to represent the learned behaviours of an experienced driver [2,3]; however, physiological limitations faced by all human drivers have been ignored.

A substantial amount of research has been carried out in the fields of biology and neuroscience to understand human perception, cognition and action in various control tasks, reviewed in detail in [4]. In many sensorimotor tasks, humans develop an internal model of their surroundings [5,6]. This allows measurements from different senses to be integrated in a statistically optimal fashion [7–10] and used to plan an appropriate control

CONTACT C. J. Nash  cn320@cam.ac.uk

 Supplemental data for this article can be accessed here. <https://doi.org/10.1080/00423114.2017.1326615>

© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

action. The main sensory systems used by drivers are the visual [11–13], vestibular [14–17] and somatosensory systems [18–21]. Cognitive responses to the signals measured by these systems are delayed due to latencies in sensor responses [22–24], nerve conduction [25] and neural processing [26]. Human sensory perception is also limited by thresholds below which sensory measurements are not perceived [27–29] and the ‘just noticeable difference’ between two similar sensory stimuli which increases linearly with stimulus amplitude [30,31]. Soyka et al. [32,33] showed that sensory thresholds can be explained by considering the signal amplitude required to exceed the level of random background noise, caused by sensory limitations and spontaneous neuron firing in the brain. Studies have shown that thresholds are larger during an active control task involving multiple stimuli compared with the passive, unimodal measurements presented in the literature [34–36], therefore sensory noise characteristics for active tasks cannot be inferred directly from passive threshold measurements. Recent studies in the aerospace industry have shown how system identification methods can be used to gather information about pilots’ use of sensory information during an active control task [37–39].

A new linear model of driver steering control was introduced in [40], incorporating models of the driver’s sensory dynamics and limitations. In [41] the model was used to explain measured pilot control behaviour from [38], and in [42] the model was updated to account for potential sensory conflicts between physical and visual motion when experiments are performed in a driving simulator. Experiments were carried out in a simulator with human drivers, adjusting the physical motion so that the separate roles of the driver’s visual and vestibular systems could be studied. A parametric identification procedure was used to find optimal model parameter values, resulting in a good fit to measured steering responses. The derivation of the linear driver model developed in [40,42] is presented in Section 2.

A linear model is a reasonable representation of normal driving conditions, however, in more extreme manoeuvres the vehicle may operate near the friction limit of the tyres and the operating point of the vehicle may vary rapidly. It is therefore necessary to develop a model which can describe a driver’s control of nonlinear vehicle dynamics. Control of nonlinear vehicles has previously been motivated by finding the minimum lap time for a racing car [43–45]. Nonlinear control algorithms are also increasingly being developed for autonomous or semi-autonomous active steering systems [46–48]. However, these applications involve finding an optimal control performance rather than matching the behaviour of a human driver. Various studies have attempted to model the nonlinear control behaviour of a human driver; however, they have neglected the effects of sensory dynamics [49–51]. The linear driver model is therefore extended in Section 3 to give a more realistic nonlinear driver model that takes account of the limitations of a human driver, and parameter values for the new model are given in Section 4.

Due to the complexity of optimal control and state estimation for a nonlinear system, a variety of approaches have previously been suggested involving different simplifications and assumptions [43–52]. The aim of the work presented in this paper is to implement and compare these methods, and evaluate their effectiveness for modelling human steering control behaviour. The most appropriate combinations will be tested against the behaviour of human drivers in future work. Simulations carried out using the model to compare different nonlinear control and state estimation methods are described in Section 5. The results of these simulations are presented in Section 6 and discussed in Section 7. The main

conclusions are summarised in Section 8. This work extends preliminary results presented in [53].

2. Linear driver model

Vehicle steering control can be represented as a combined target-following and disturbance-rejection task [54], as shown in Figure 1. The driver must follow a target path of curvature f_κ while reacting to disturbances f_v and f_ω on the lateral and angular velocities of the vehicle. The target-following component is generally a feedforward control task, as the driver can ‘preview’ the upcoming target path in order to plan future steering control inputs. However, in previous work experiments were carried out where the driver could not preview the upcoming target, allowing delays in the driver’s visual system to be measured [40,42]. The disturbance-rejection component is a feedback task, as the driver cannot perceive disturbances until after they have occurred.

The structure of the new driver-vehicle control model is shown in Figure 2. The model uses optimal control theory to represent a driver who has learned to steer as well as possible on average based on the information available to them. It consists of three main components: a plant, a controller and a state estimator. The dynamics of the vehicle and the driver are described by the plant, which is perturbed by target and disturbance white noise representations w_κ , w_v and w_ω , and process noise w . The plant equations are derived in Section 2.1. The plant outputs y represent the outputs of the driver’s sensory systems, and these are perturbed by measurement noise v . A state estimator calculates an estimate \hat{x} of the plant states, using an internal model of the plant to reduce the uncertainty in the noisy sensory measurements. The state estimator is described in Section 2.2. Finally, a controller uses this state estimate and the internal model to calculate an optimal plant input $\hat{\delta}$, as discussed in Section 2.3.

2.1. Plant

The plant describes the open-loop dynamics of the driver–vehicle system. In addition, the driver has an internal model of the plant which is used by the controller and state estimator [5,6]. A block diagram of the plant is shown in Figure 3. The plant is implemented as a state-space system, in discrete time with sample period T_s so that delays can be modelled explicitly. All continuous transfer functions $H(s)$ are converted to discrete state-space matrices (**A**, **B**, **C**, **D**) with states x . Discretisation is carried out using a zero-order hold

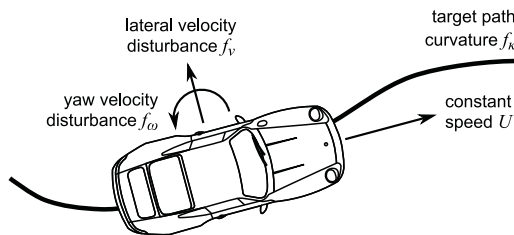


Figure 1. Representation of steering control task in new driver model, adapted from [40]

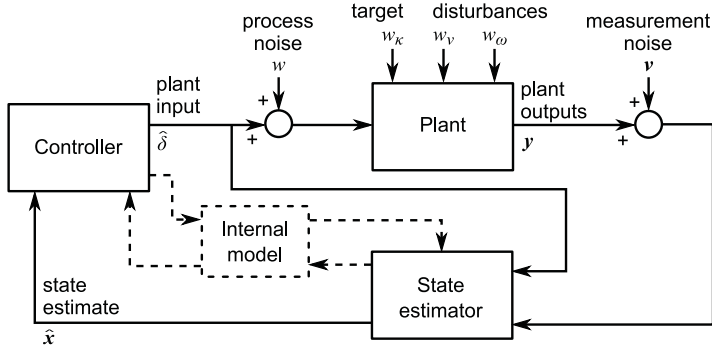


Figure 2. Overall structure of new driver model.

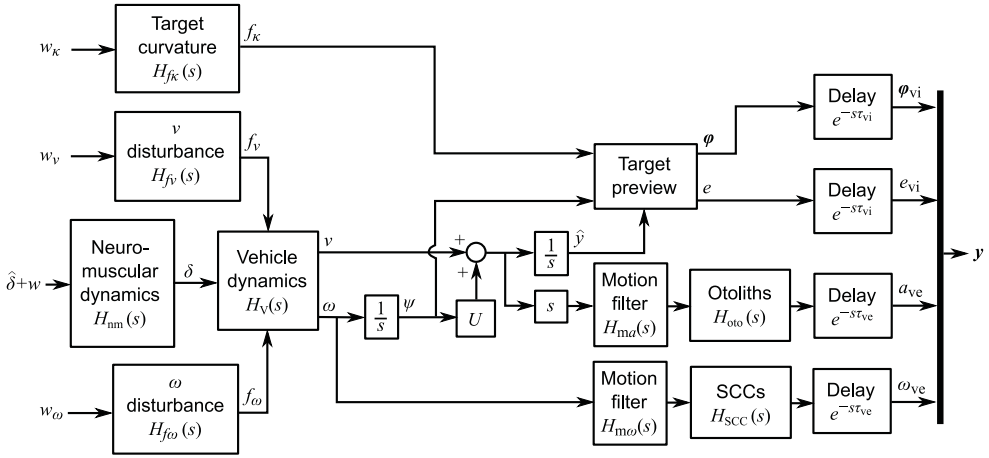


Figure 3. Block diagram of plant describing the dynamics controlled by the driver, adapted from [42].

method. In some cases this is approximated in the form $\mathbf{A} = \mathbf{I} + T_s \mathbf{A}_c$, $\mathbf{B} = T_s \mathbf{B}_c$, where \mathbf{A}_c and \mathbf{B}_c are continuous state-space matrices.

The driver's neuromuscular dynamics $H_{nm}(s)$ shape the plant input $\hat{\delta}$ (plus process noise w) to give the steering angle δ which is included as a plant state:

$$\begin{aligned} \mathbf{x}_{nm}(k+1) &= \mathbf{A}_{nm} \mathbf{x}_{nm}(k) + \mathbf{B}_{nm}(\hat{\delta}(k) + w(k)), \\ \delta(k+1) &= \mathbf{C}_{nm} \mathbf{x}_{nm}(k) + \mathbf{D}_{nm}(\hat{\delta}(k) + w(k)). \end{aligned} \quad (1)$$

Target and disturbance forcing function filters $H_{f_K}(s)$, $H_{f_v}(s)$ and $H_{f_\omega}(s)$ are written in discrete state-space form. The target curvature f_K is given by

$$\begin{aligned} \mathbf{x}_{f_K}(k+1) &= \mathbf{A}_{f_K} \mathbf{x}_{f_K}(k) + \mathbf{B}_{f_K} w_K(k), \\ f_K(k) &= \mathbf{C}_{f_K} \mathbf{x}_{f_K}(k) + \mathbf{D}_{f_K} w_K(k), \end{aligned} \quad (2)$$

and the lateral velocity disturbance f_v and angular velocity disturbance f_ω are found similarly, where w_K , w_v and w_ω are Gaussian white noise inputs with RMS magnitudes W_K , W_v and W_ω .

The vehicle $H_V(s)$ takes as inputs the steering angle δ and the disturbances f_v and f_ω , and its outputs are the vehicle lateral velocity v and angular velocity ω :

$$\begin{aligned} \mathbf{x}_V(k+1) &= \mathbf{A}_V \mathbf{x}_V(k) + \mathbf{B}_V \{\delta(k) \ f_v(k) \ f_\omega(k)\}^T, \\ \{v(k) \ \omega(k)\}^T &= \mathbf{C}_V \mathbf{x}_V(k) + \mathbf{D}_V \{\delta(k) \ f_v(k) \ f_\omega(k)\}^T, \end{aligned} \quad (3)$$

which expands to

$$\begin{aligned} \mathbf{x}_V(k+1) &= \mathbf{A}_V \mathbf{x}_V(k) + \mathbf{B}_{V(:,1)} \delta(k) + \mathbf{B}_{V(:,2)} \mathbf{C}_{fv} \mathbf{x}_{fv}(k) \\ &\quad + \mathbf{B}_{V(:,2)} \mathbf{D}_{fv} w_v(k) + \mathbf{B}_{V(:,3)} \mathbf{C}_{f\omega} \mathbf{x}_{f\omega}(k) + \mathbf{B}_{V(:,3)} \mathbf{D}_{f\omega} w_\omega(k), \\ v(k) &= \mathbf{C}_{V(1,:)} \mathbf{x}_V(k) + \mathbf{D}_{V(1,1)} \delta(k) + \mathbf{D}_{V(1,2)} \mathbf{C}_{fv} \mathbf{x}_{fv}(k) \\ &\quad + \mathbf{D}_{V(1,2)} \mathbf{D}_{fv} w_v(k) + \mathbf{D}_{V(1,3)} \mathbf{C}_{f\omega} \mathbf{x}_{f\omega}(k) + \mathbf{D}_{V(1,3)} \mathbf{D}_{f\omega} w_\omega(k), \\ \omega(k) &= \mathbf{C}_{V(2,:)} \mathbf{x}_V(k) + \mathbf{D}_{V(2,1)} \delta(k) + \mathbf{D}_{V(2,2)} \mathbf{C}_{fv} \mathbf{x}_{fv}(k) \\ &\quad + \mathbf{D}_{V(2,2)} \mathbf{D}_{fv} w_v(k) + \mathbf{D}_{V(2,3)} \mathbf{C}_{f\omega} \mathbf{x}_{f\omega}(k) + \mathbf{D}_{V(2,3)} \mathbf{D}_{f\omega} w_\omega(k), \end{aligned} \quad (4)$$

where $\mathbf{M}_{(i,j)}$ indicates the i th row and the j th column of matrix \mathbf{M} , and ‘:’ represents the entire row or column.

The plant outputs are subjected to visual and vestibular delays τ_{vi} and τ_{ve} , consisting of $N_{vi} = \tau_{vi}/T_s$ and $N_{ve} = \tau_{ve}/T_s$ time steps. The number of states required can be reduced by applying the delays to v and ω instead, then calculating the plant outputs from these delayed values. Delayed values of v are found using a shift register of length $N_m = \max(N_{vi}, N_{ve})$:

$$\begin{aligned} \begin{Bmatrix} v(k) \\ \vdots \\ v(k - N_m + 1) \end{Bmatrix} &= \begin{bmatrix} \mathbf{0}_{[1, N_m-1]} & 0 \\ \mathbf{I}_{[N_m-1, N_m-1]} & \mathbf{0}_{[N_m-1, 1]} \end{bmatrix} \begin{Bmatrix} v(k-1) \\ \vdots \\ v(k - N_m) \end{Bmatrix} + \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} v(k), \\ \mathbf{x}_{\tau v}(k+1) &= \mathbf{A}_\tau \mathbf{x}_{\tau v}(k) + \mathbf{B}_\tau v(k) \\ &= \mathbf{A}_\tau \mathbf{x}_{\tau v}(k) + \mathbf{B}_\tau \mathbf{C}_{V(1,:)} \mathbf{x}_V(k) + \mathbf{B}_\tau \mathbf{D}_{V(1,1)} \delta(k) \\ &\quad + \mathbf{B}_\tau \mathbf{D}_{V(1,2)} \mathbf{C}_{fv} \mathbf{x}_{fv}(k) + \mathbf{B}_\tau \mathbf{D}_{V(1,2)} \mathbf{D}_{fv} w_v(k) \\ &\quad + \mathbf{B}_\tau \mathbf{D}_{V(1,3)} \mathbf{C}_{f\omega} \mathbf{x}_{f\omega}(k) + \mathbf{B}_\tau \mathbf{D}_{V(1,3)} \mathbf{D}_{f\omega} w_\omega(k), \end{aligned} \quad (5)$$

where \mathbf{I} is the identity matrix, $\mathbf{0}$ is a matrix of zeros, and $\mathbf{M}_{[i,j]}$ is a matrix with i rows and j columns. Delayed values of ω are found similarly:

$$\begin{aligned} \mathbf{x}_{\tau \omega}(k+1) &= \mathbf{A}_\tau \mathbf{x}_{\tau \omega}(k) + \mathbf{B}_\tau \mathbf{C}_{V(2,:)} \mathbf{x}_V(k) + \mathbf{B}_\tau \mathbf{D}_{V(2,1)} \delta(k) \\ &\quad + \mathbf{B}_\tau \mathbf{D}_{V(2,2)} \mathbf{C}_{fv} \mathbf{x}_{fv}(k) + \mathbf{B}_\tau \mathbf{D}_{V(2,2)} \mathbf{D}_{fv} w_v(k) \\ &\quad + \mathbf{B}_\tau \mathbf{D}_{V(2,3)} \mathbf{C}_{f\omega} \mathbf{x}_{f\omega}(k) + \mathbf{B}_\tau \mathbf{D}_{V(2,3)} \mathbf{D}_{f\omega} w_\omega(k) \end{aligned} \quad (6)$$

The visual system measurements depend on ω , delayed by τ_{vi} . This is given by

$$\omega(k - N_{vi}) = [\mathbf{0}_{[1, N_{vi}-1]} \ 1 \ \mathbf{0}_{[1, N_m - N_{vi}]}] \mathbf{x}_{\tau \omega}(k) = \mathbf{C}_{\tau vi} \mathbf{x}_{\tau \omega}(k). \quad (7)$$

Similarly, the semi-circular canals (SCCs) take measurements of ω , delayed by τ_{ve} :

$$\omega(k - N_{ve}) = [\mathbf{0}_{[1, N_{ve}-1]} \ 1 \ \mathbf{0}_{[1, N_m - N_{ve}]}] \mathbf{x}_{\tau \omega}(k) = \mathbf{C}_{\tau \omega} \mathbf{x}_{\tau \omega}(k) \quad (8)$$

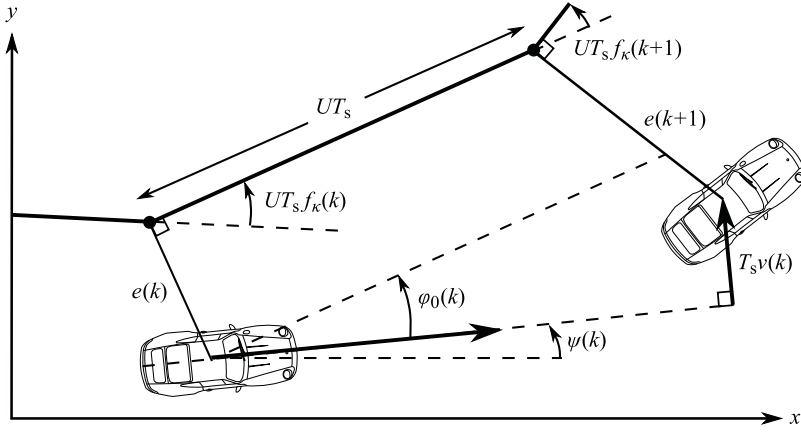


Figure 4. Visual system measurements: Path-following error e and previewed road angles ϕ .

The otoliths measure the vehicle's lateral acceleration, which is found from:

$$a = \frac{dv}{dt} + U\omega \quad (9)$$

In discrete time, and delayed by τ_{ve} , this becomes

$$\begin{aligned} a(k - N_{ve}) &= \frac{1}{T_s} v(k - N_{ve} + 1) - \frac{1}{T_s} v(k - N_{ve}) + U\omega(k - N_{ve}) \\ &= \begin{bmatrix} \mathbf{0}_{[1, N_{ve}-2]} & \frac{1}{T_s} & \frac{-1}{T_s} & \mathbf{0}_{[1, N_m - N_{ve}]} \end{bmatrix} \mathbf{x}_{\tau v}(k) + U\mathbf{C}_{\tau\omega} \mathbf{x}_{\tau\omega}(k) \\ &= \mathbf{C}_{\tau v} \mathbf{x}_{\tau v}(k) + U\mathbf{C}_{\tau\omega} \mathbf{x}_{\tau\omega}(k). \end{aligned} \quad (10)$$

The visual system of the driver measures the path-following error e and previews upcoming target angles $\phi = [\phi_0(k) \dots \phi_{N_p}(k)]$ relative to the vehicle, as shown in Figure 4. This gives a prediction horizon of $N_p = T_p/T_s$ time steps. The target angles are stored as states in a shift register, given by

$$\phi_n(k+1) = \phi_{(n+1)}(k) - T_s\omega(k) \quad (11)$$

with the most distant previewed angle ϕ_{N_p} given by

$$\phi_{N_p}(k+1) = \phi_{N_p}(k) - T_s\omega(k) + UT_s f_k(k + N_p), \quad (12)$$

In matrix form, and delayed by τ_{vi} :

$$\begin{bmatrix} \tilde{\mathbf{x}}_{f_k}(k+1) \\ \phi_{N_p}(k - N_{vi} + 1) \\ \vdots \\ \phi_0(k - N_{vi} + 1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{f_k} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ UT_s \mathbf{C}_{f_k} & 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{f_k}(k) \\ \phi_{N_p}(k - N_{vi}) \\ \vdots \\ \phi_0(k - N_{vi}) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{f_k} \\ UT_s \mathbf{D}_{f_k} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tilde{\mathbf{w}}_k(k)$$

$$\begin{aligned}
& + \begin{bmatrix} \mathbf{0} \\ -T_s \mathbf{1}_{[N_p+1, 1]} \end{bmatrix} \omega(k - N_{vi}), \\
\phi_{vi}(k) = & \begin{Bmatrix} \phi_{N_p}(k - N_{vi}) \\ \vdots \\ \phi_0(k - N_{vi}) \end{Bmatrix} = [\mathbf{0} \quad \mathbf{I}_{[N_p+1, N_p+1]}] \mathbf{x}_\phi(k),
\end{aligned} \tag{13}$$

where $\tilde{w}_\kappa(k) = w_\kappa(k + N_p - N_{vi})$, $\tilde{\mathbf{x}}_{f\kappa}(k) = \mathbf{x}_{f\kappa}(k + N_p - N_{vi})$ and $\mathbf{1}$ is a matrix of ones. This can be written more concisely as

$$\begin{aligned}
\mathbf{x}_\phi(k+1) &= \mathbf{A}_\phi \mathbf{x}_\phi(k) + \mathbf{B}_\phi \tilde{w}_\kappa(k) + \mathbf{B}_{\phi\omega} \mathbf{C}_{\tau vi} \mathbf{x}_{\tau\omega}(k), \\
\phi_{vi}(k) &= \mathbf{C}_\phi \mathbf{x}_\phi(k).
\end{aligned} \tag{14}$$

Assuming ϕ_0 and f_κ are small, from the geometry in Figure 4 the path-following error e can be found from:

$$e(k+1) = e(k) + UT_s \phi_0(k) - T_s v(k), \tag{15}$$

which when delayed by τ_{vi} becomes

$$\begin{aligned}
e_{vi}(k+1) &= e(k - N_{vi} + 1) = e(k - N_{vi}) + UT_s \phi_0(k - N_{vi}) - T_s v(k - N_{vi}) \\
&= e(k - N_{vi}) + UT_s [\mathbf{0} \quad \mathbf{1}] \mathbf{x}_\phi(k) - T_s \mathbf{C}_{\tau vi} \mathbf{x}_{\tau v}(k).
\end{aligned} \tag{16}$$

The transfer function between the delayed angular velocity and the angular velocity ω_{ve} perceived by the SCCs is $H_\omega(s) = H_{\text{SCC}}(s)H_{m\omega}(s)$, where $H_{\text{SCC}}(s)$ is a transfer function representing the dynamics of the SCCs and $H_{m\omega}(s)$ is the scaling or filtering applied to the yaw motion. The perceived angular velocity is then found from:

$$\begin{aligned}
\mathbf{x}_\omega(k+1) &= \mathbf{A}_\omega \mathbf{x}_\omega(k) + \mathbf{B}_\omega \omega(k - N_{ve}) = \mathbf{A}_\omega \mathbf{x}_\omega(k) + \mathbf{B}_\omega \mathbf{C}_{\tau\omega} \mathbf{x}_{\tau\omega}(k), \\
\omega_{ve}(k) &= \mathbf{C}_\omega \mathbf{x}_\omega(k) + \mathbf{D}_\omega \omega(k - N_{ve}) = \mathbf{C}_\omega \mathbf{x}_\omega(k) + \mathbf{D}_\omega \mathbf{C}_{\tau\omega} \mathbf{x}_{\tau\omega}(k).
\end{aligned} \tag{17}$$

The transfer function between the delayed lateral acceleration and the acceleration a_{ve} perceived by the otoliths is $H_a(s) = H_{\text{oto}}(s)H_{ma}(s)$, where $H_{\text{oto}}(s)$ is a transfer function representing the dynamics of the otoliths and $H_{ma}(s)$ is the scaling or filtering applied to the lateral acceleration. The perceived lateral acceleration is then found from:

$$\begin{aligned}
\mathbf{x}_a(k+1) &= \mathbf{A}_a \mathbf{x}_a(k) + \mathbf{B}_a a(k - N_{ve}) = \mathbf{A}_a \mathbf{x}_a(k) + \mathbf{B}_a \mathbf{C}_{\tau v} \mathbf{x}_{\tau v}(k) + U \mathbf{B}_a \mathbf{C}_{\tau\omega} \mathbf{x}_{\tau\omega}(k), \\
a_{ve}(k) &= \mathbf{C}_a \mathbf{x}_a(k) + \mathbf{D}_a a(k - N_{ve}) = \mathbf{C}_a \mathbf{x}_a(k) + \mathbf{D}_a \mathbf{C}_{\tau v} \mathbf{x}_{\tau v}(k) + U \mathbf{D}_a \mathbf{C}_{\tau\omega} \mathbf{x}_{\tau\omega}(k).
\end{aligned} \tag{18}$$

Combining all these equations gives the complete plant:

$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A} \mathbf{x}(k) + \mathbf{B} \hat{\delta}(k) + [\mathbf{B} \quad \mathbf{G}_v \quad \mathbf{G}_\omega \quad \mathbf{G}_\kappa] \{w(k) \quad w_v(k) \quad w_\omega(k) \quad \tilde{w}_\kappa(k)\}^T, \\
\mathbf{y}(k) &= \mathbf{C} \mathbf{x}(k),
\end{aligned} \tag{19}$$

where

$$\mathbf{x} = \{\mathbf{x}_V \quad \delta \quad \mathbf{x}_{nm} \quad \mathbf{x}_{fv} \quad \mathbf{x}_{f\omega} \quad \mathbf{x}_{\tau v} \quad \mathbf{x}_{\tau\omega} \quad \mathbf{x}_\phi \quad \mathbf{x}_a \quad \mathbf{x}_\omega \quad e_{vi}\}^T,$$

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} \mathbf{A}_v & \mathbf{B}_{V(:,1)} & \mathbf{0} & \mathbf{B}_{V(:,2)}\mathbf{C}_{fv} & \mathbf{B}_{V(:,3)}\mathbf{C}_{f\omega} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{nm} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{nm} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{fv} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{f\omega} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_\tau\mathbf{C}_{V(1,:)} & \mathbf{B}_\tau\mathbf{D}_{V(1,1)} & \mathbf{0} & \mathbf{B}_\tau\mathbf{D}_{V(1,2)}\mathbf{C}_{fv} & \mathbf{B}_\tau\mathbf{D}_{V(1,3)}\mathbf{C}_{f\omega} & \mathbf{A}_\tau & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_\tau\mathbf{C}_{V(2,:)} & \mathbf{B}_\tau\mathbf{D}_{V(2,1)} & \mathbf{0} & \mathbf{B}_\tau\mathbf{D}_{V(2,2)}\mathbf{C}_{fv} & \mathbf{B}_\tau\mathbf{D}_{V(2,3)}\mathbf{C}_{f\omega} & \mathbf{0} & \mathbf{A}_\tau & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{\phi\omega}\mathbf{C}_{\tau vi} & \mathbf{A}_\phi & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_a\mathbf{C}_{\tau v} & \mathbf{UB}_a\mathbf{C}_{\tau\omega} & \mathbf{0} & \mathbf{A}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_\omega\mathbf{C}_{\tau\omega} & \mathbf{0} & \mathbf{0} & \mathbf{A}_\omega \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -T_s\mathbf{C}_{\tau vi} & \mathbf{0} & [\mathbf{0} \ UT_s] & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}, \\
\mathbf{B} &= [\mathbf{0} \ \mathbf{D}_{nm} \ \mathbf{B}_{nm} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0}]^T, \\
\mathbf{G}_v &= [\mathbf{B}_{V(:,2)}\mathbf{D}_{fv} \ \mathbf{0} \ \mathbf{0} \ \mathbf{B}_{fv} \ \mathbf{0} \ \mathbf{B}_\tau\mathbf{D}_{V(1,2)}\mathbf{D}_{fv} \ \mathbf{B}_\tau\mathbf{D}_{V(2,2)}\mathbf{D}_{fv} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0}]^T, \\
\mathbf{G}_\omega &= [\mathbf{B}_{V(:,3)}\mathbf{D}_{f\omega} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{B}_{f\omega} \ \mathbf{B}_\tau\mathbf{D}_{V(1,3)}\mathbf{D}_{f\omega} \ \mathbf{B}_\tau\mathbf{D}_{V(2,3)}\mathbf{D}_{f\omega} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0}]^T, \\
\mathbf{G}_\kappa &= [\mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{0} \ \mathbf{B}_\phi \ \mathbf{0} \ \mathbf{0} \ \mathbf{0}]^T, \\
\mathbf{y} &= \{\phi_{vi} \ e_{vi} \ a_{ve} \ \omega_{ve}\}^T, \\
\mathbf{C} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}_\phi & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_a\mathbf{C}_{\tau v} & \mathbf{UD}_a\mathbf{C}_{\tau\omega} & \mathbf{0} & \mathbf{C}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_\omega\mathbf{C}_{\tau\omega} & \mathbf{0} & \mathbf{0} & \mathbf{C}_\omega & \mathbf{0} \end{bmatrix}. \tag{20}
\end{aligned}$$

The small angle assumptions made in this derivation allow a linear simulation to be run even with large overall angles in the target path and the vehicle dynamics. However, the Cartesian coordinates of the target path and vehicle must be calculated. For the target, the x and y components x_t and y_t are given by

$$\begin{aligned}
x_t(k+1) &= x_t(k) + UT_s \cos\left(\sum_{i=0}^k UT_s f_k(i)\right), \\
y_t(k+1) &= y_t(k) + UT_s \sin\left(\sum_{i=0}^k UT_s f_k(i)\right). \tag{21}
\end{aligned}$$

The position of the vehicle is linearised about the target path as shown in Figure 4 [43], and is given by finding the point located e away from the target coordinate in a perpendicular direction:

$$\begin{aligned}
x(k) &= x_t(k) + e(k) \sin\left(\sum_{i=0}^k UT_s f_k(i)\right), \\
y(k) &= y_t(k) - e(k) \cos\left(\sum_{i=0}^k UT_s f_k(i)\right). \tag{22}
\end{aligned}$$

2.2. State estimator

The controller requires the full plant state vector \mathbf{x} in order to calculate the optimal plant input. However, the driver does not have access to all the states, and only measures the

outputs \mathbf{y} of the plant perturbed by measurement noise \mathbf{v} . An estimate $\hat{\mathbf{x}}$ of the plant states is obtained using a Kalman filter [10], based on an internal model of the plant, a measurement of the noise-free plant input $\hat{\delta}$ and noisy measurements $(\mathbf{y} + \mathbf{v})$ of the outputs. The noise covariance matrices \mathbf{Q}_{KF} and \mathbf{R}_{KF} are given by

$$\begin{aligned}\mathbf{Q}_{\text{KF}} &= \text{diag} \left(\begin{bmatrix} W^2 & W_v^2 & W_\omega^2 & W_\kappa^2 \end{bmatrix} \right), \\ \mathbf{R}_{\text{KF}} &= \text{diag} \left(\begin{bmatrix} V_\phi^2 \mathbf{1}_{[1, N_p+1]} & V_e^2 & V_a^2 & V_\omega^2 \end{bmatrix} \right),\end{aligned}\quad (23)$$

where W^2 is the variance of the process noise w and V_ϕ^2 , V_e^2 , V_a^2 , and V_ω^2 are the variances of the measurement noise added to the plant outputs ϕ_{vi} , e_{vi} , a_{ve} , and ω_{ve} . In some implementations, \mathbf{Q}_{KF} is replaced by the covariance of the process noise on the states, given by

$$\hat{\mathbf{Q}}_{\text{KF}} = [\mathbf{B} \ \mathbf{G}_v \ \mathbf{G}_\omega \ \mathbf{G}_\kappa] \mathbf{Q}_{\text{KF}} [\mathbf{B} \ \mathbf{G}_v \ \mathbf{G}_\omega \ \mathbf{G}_\kappa]^T \quad (24)$$

The state estimate is given by

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\hat{\delta}(k) + \mathbf{L}(\mathbf{y}(k) + \mathbf{v}(k) - \mathbf{C}\hat{\mathbf{x}}(k)), \quad (25)$$

where

$$\mathbf{L} = \mathbf{A}\mathbf{P}\mathbf{C}^T(\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}_{\text{KF}})^{-1} \quad (26)$$

and \mathbf{P} solves the discrete Riccati equation:

$$\mathbf{A}^T\mathbf{P}\mathbf{A} - \mathbf{P} - \mathbf{A}^T\mathbf{P}\mathbf{C}(\mathbf{C}^T\mathbf{P}\mathbf{C} + \mathbf{R}_{\text{KF}})^{-1}\mathbf{C}^T\mathbf{P}\mathbf{A} + \hat{\mathbf{Q}}_{\text{KF}} = \mathbf{0}. \quad (27)$$

This gives a linear time-invariant Kalman filter $\mathbf{H}_{\text{KF}}(s)$, which can be calculated using the Matlab function *kalman*. The state estimates $\hat{\mathbf{x}}$ can then be found from:

$$\hat{\mathbf{x}}(s) = \mathbf{H}_{\text{KF}}(s) \begin{Bmatrix} \hat{\delta}(s) \\ \mathbf{y}(s) + \mathbf{v}(s) \end{Bmatrix}. \quad (28)$$

2.3. Controller

An optimal controller is used to minimise a cost function weighting the path-following error against steering effort. There are two main implementations of optimal control, a linear quadratic regulator (LQR) and model predictive control (MPC), which have been found to be identical when used under the same conditions [3]. For the linear plant, an LQR controller was used as it can find the optimum control input over an infinite control horizon, assuming white noise inputs beyond the previewed target path.

The LQR method involves calculating an optimal gain vector \mathbf{K}_{LQ} , which acts on the plant states to give an optimal plant input $\hat{\delta}$ which minimises a cost function J . Additive white noise does not affect the optimal solution, so the white noise inputs w , w_κ , w_v and w_ω can be ignored. The cost function incorporates costs on the tracking error e_{vi} and the plant input $\hat{\delta}$, weighted by q_e and q_δ :

$$J = \sum_{k=0}^{\infty} \{q_e e_{vi}(k)^2 + q_\delta \hat{\delta}(k)^2\}$$

$$= \sum_{k=0}^{\infty} \{ \mathbf{x}(k)^T \mathbf{Q}_{LQ} \mathbf{x}(k) + \hat{\delta}(k)^T \mathbf{R}_{LQ} \hat{\delta}(k) \}, \quad (29)$$

where

$$\mathbf{Q}_{LQ} = \text{diag}([0 \quad \cdots \quad 0 \quad q_e]), \quad \mathbf{R}_{LQ} = q_{\delta}. \quad (30)$$

The optimal plant input is [3]:

$$\hat{\delta}(k) = -\mathbf{K}_{LQ} \mathbf{x}(k), \quad (31)$$

where

$$\mathbf{K}_{LQ} = (\mathbf{B}^T \mathbf{S} \mathbf{B} + \mathbf{R}_{LQ})^{-1} \mathbf{B}^T \mathbf{S} \mathbf{A} \quad (32)$$

and \mathbf{S} solves the discrete Riccati equation:

$$\mathbf{A}^T \mathbf{S} \mathbf{A} - \mathbf{S} - \mathbf{A}^T \mathbf{S} \mathbf{B} (\mathbf{B}^T \mathbf{S} \mathbf{B} + \mathbf{R}_{LQ})^{-1} \mathbf{B}^T \mathbf{S} \mathbf{A} + \mathbf{Q}_{LQ} = \mathbf{0}. \quad (33)$$

The optimal gain \mathbf{K}_{LQ} can be found in this way using the Matlab function *dlqr*.

3. Nonlinear driver model

In this section the linear driver model derived in Section 2 is extended to control a constant speed vehicle with nonlinear tyres. Nonlinear optimal control and state estimation are much more difficult than their linear equivalents. However, it is possible to simplify the task by approximating the system dynamics. There are various ways in which this can be done, therefore, several state estimators and controllers are implemented. The performance of the different methods is compared in Section 6 using simulations described in Section 5.

3.1. Plant

The only difference between the linear and nonlinear plant is a nonlinear vehicle. The disturbances are added using the linear relationship in Equation (3); however, the dependence on the steering angle and previous states is nonlinear. The vehicle states (v and ω) are therefore found from:

$$\begin{Bmatrix} v(k+1) \\ \omega(k+1) \end{Bmatrix} = A_V \left(\begin{Bmatrix} v(k) \\ \omega(k) \end{Bmatrix}, \delta(k) \right) + \mathbf{B}_{V(:,2)} \mathbf{C}_{fv} \mathbf{x}_{fv} + \mathbf{B}_{V(:,3)} \mathbf{C}_{f\omega} \mathbf{x}_{f\omega}, \quad (34)$$

where A_V is a nonlinear function. The remaining states are calculated as for the linear model using Equation (19), and the same linear output equation is used.

In order to reduce the computational load of the nonlinear controllers, a reduced plant is defined which includes only the states required by the controller. It should be noted that the state estimator still requires the complete plant. The controller minimises a cost function weighting $\hat{\delta}$ against e , therefore any parts of the plant which do not feed into e in Figure 3 are ignored. This includes the motion filters, sensory dynamics and delays. Furthermore, using an MPC formulation allows the target path to be implemented as a

reference without including the preview shift register in the reduced plant. The driver's visual system previews $(N_p + 1)$ path angles; however, due to the visual delay N_{vi} of the measured angles are already behind the current position of the vehicle. Therefore, the controllers described in Section 3.3 evaluate the reduced plant equations from the current time step k to $(k + N_p - N_{vi} + 1)$.

Equation (11) can be adapted to find future values of ϕ_0 from the measured values of ϕ_n :

$$\phi_0(k + n) = \phi_n(k) - \hat{\psi}(k + n) \quad (35)$$

where $\hat{\psi}$ is the yaw angle of the vehicle, referenced to the current yaw angle so that $\hat{\psi}(k) = 0$. It can be found from:

$$\begin{aligned} \hat{\psi}(k + n + 1) &= \hat{\psi}(k + n) + T_s \omega(k + n) \\ &= \hat{\psi}(k + n) + T_s \mathbf{C}_{V(2,:)} \mathbf{x}_V(k + n) + T_s \mathbf{D}_{V(2,1)} \delta(k + n) \\ &\quad + T_s \mathbf{D}_{V(2,2)} \mathbf{C}_{fv} \mathbf{x}_{fv}(k + n) + T_s \mathbf{D}_{V(2,2)} \mathbf{D}_{fv} w_v(k + n) \\ &\quad + T_s \mathbf{D}_{V(2,3)} \mathbf{C}_{f\omega} \mathbf{x}_{f\omega}(k + n) + T_s \mathbf{D}_{V(2,3)} \mathbf{D}_{f\omega} w_\omega(k + n). \end{aligned} \quad (36)$$

Equation (15) can then be rewritten to separate the predicted e over the prediction horizon into a controllable part \hat{y} and a reference \hat{y}_{ref} :

$$\begin{aligned} e(k + n + 1) &= e(k + n) + UT_s \phi_0(k + n) - T_s v(k + n) \\ &= e(k + n) + UT_s \phi_n(k) - UT_s \hat{\psi}(k + n) - T_s v(k + n) \\ &= \hat{y}_{\text{ref}}(k + n + 1) - \hat{y}(k + n + 1). \end{aligned} \quad (37)$$

The reference trajectory is found from:

$$\hat{y}_{\text{ref}}(k + n) = \sum_{i=0}^{n-1} (UT_s \phi_i(k)) \quad (38)$$

and the controllable part is

$$\begin{aligned} \hat{y}(k + n + 1) &= \hat{y}(k + n) + T_s v(k + n) + UT_s \psi(k + n) \\ &= \hat{y}(k + n) + T_s \mathbf{C}_{V(1,:)} \mathbf{x}_V(k + n) + T_s \mathbf{D}_{V(1,1)} \delta(k + n) \\ &\quad + T_s \mathbf{D}_{V(1,2)} \mathbf{C}_{fv} \mathbf{x}_{fv}(k + n) + T_s \mathbf{D}_{V(1,2)} \mathbf{D}_{fv} w_v(k + n) \\ &\quad + T_s \mathbf{D}_{V(1,3)} \mathbf{C}_{f\omega} \mathbf{x}_{f\omega}(k + n) + T_s \mathbf{D}_{V(1,3)} \mathbf{D}_{f\omega} w_\omega(k + n) + UT_s \hat{\psi}(k + n). \end{aligned} \quad (39)$$

If the target is close to a straight line, \hat{y} and \hat{y}_{ref} represent the vehicle and target lateral displacements.

The complete reduced plant for a linear vehicle has the same form as Equation (19), with state-space matrices:

$$\begin{aligned}
 \mathbf{x}_R &= \{x_V \quad \delta \quad x_{nm} \quad x_{fv} \quad x_{f\omega} \quad \hat{\psi} \quad \hat{y}\}^T, \\
 \mathbf{A}_R &= \begin{bmatrix} \mathbf{A}_V & \mathbf{B}_{V(:,1)} & \mathbf{0} & \mathbf{B}_{V(:,2)}\mathbf{C}_{fv} & \mathbf{B}_{V(:,3)}\mathbf{C}_{f\omega} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{C}_{nm} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{nm} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{fv} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{f\omega} & \mathbf{0} & \mathbf{0} \\ T_s\mathbf{C}_{V(2,:)} & T_s\mathbf{D}_{V(2,1)} & \mathbf{0} & T_s\mathbf{D}_{V(2,2)}\mathbf{C}_{fv} & T_s\mathbf{D}_{V(2,3)}\mathbf{C}_{f\omega} & 1 & 0 \\ T_s\mathbf{C}_{V(1,:)} & T_s\mathbf{D}_{V(1,1)} & \mathbf{0} & T_s\mathbf{D}_{V(1,2)}\mathbf{C}_{fv} & T_s\mathbf{D}_{V(1,3)}\mathbf{C}_{f\omega} & UT_s & 1 \end{bmatrix}, \\
 \mathbf{B}_R &= [\mathbf{0} \quad \mathbf{D}_{nm} \quad \mathbf{B}_{nm} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}]^T, \\
 \mathbf{G}_{vR} &= [\mathbf{B}_{V(:,2)}\mathbf{D}_{fv} \quad 0 \quad 0 \quad \mathbf{B}_{fv} \quad 0 \quad T_s\mathbf{D}_{V(2,2)}\mathbf{D}_{fv} \quad T_s\mathbf{D}_{V(1,2)}\mathbf{D}_{fv}]^T, \\
 \mathbf{G}_{\omega R} &= [\mathbf{B}_{V(:,3)}\mathbf{D}_{f\omega} \quad 0 \quad 0 \quad 0 \quad \mathbf{B}_{f\omega} \quad T_s\mathbf{D}_{V(2,3)}\mathbf{D}_{f\omega} \quad T_s\mathbf{D}_{V(1,3)}\mathbf{D}_{f\omega}]^T, \\
 \mathbf{G}_{\kappa R} &= [\mathbf{0} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T, \\
 \mathbf{y}_R &= \hat{y}, \\
 \mathbf{C}_R &= [\mathbf{0} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1],
 \end{aligned} \tag{40}$$

although for a nonlinear vehicle the vehicle states are calculated using Equation (34).

For the control calculation at each time step, it is necessary to find the reduced state vector $\mathbf{x}_R(k)$ and reference trajectory $\hat{\mathbf{y}}_{\text{ref}}(k)$ from the full state estimate $\hat{\mathbf{x}}(k)$. Most of the states in $\mathbf{x}_R(k)$ can be taken directly from the equivalent states in $\hat{\mathbf{x}}(k)$, and the yaw angle $\hat{\psi}(k)$ is defined to start at zero. The value of $\hat{y}(k)$ is equal to the negative of the current path-following error $e(k)$. However, only the delayed version $e_{vi}(k)$ is stored directly in the full state vector. Therefore, $\hat{y}(k)$ must be calculated by iterating Equation (15) using the stored delayed values of v , ω and ϕ_0 , giving:

$$\begin{aligned}
 \hat{y}(k) &= -e_{vi}(k) - UT_s \sum_{n=0}^{N_{vi}-1} \{\phi_n(k - N_{vi})\} \\
 &\quad + T_s \sum_{n=-N_{vi}}^{-1} \{v(k+n)\} + UT_s^2 \sum_{n=-N_{vi}}^{-2} \{n\omega(k+n)\}.
 \end{aligned} \tag{41}$$

Similarly, the reference trajectory depends through Equation (38) on previewed angles $\phi_n(k)$; however, the full state vector contains delayed values $\phi_n(k - N_{vi})$. The required angles can be calculated by iterating Equation (11) over the visual delay:

$$\phi_n(k) = \phi_{(n+N_{vi})}(k - N_{vi}) - T_s \sum_{i=-N_{vi}}^{-1} \{\omega(k+i)\}. \tag{42}$$

3.2. State estimators

State estimation for a nonlinear plant can be achieved using an extended Kalman filter, which operates on the same principles as the Kalman filter described in Section 2.2. Extended Kalman filters approximate the nonlinear plant dynamics by linearisation or similar transformations. Four variations of the extended Kalman filter are implemented using code provided by the EKF/UKF Matlab toolbox [52]. These four extended Kalman filters are also compared with full state feedback, giving five state estimators in total:

- LKF: Linear Kalman filter: A time-varying implementation of a linear Kalman filter, found by linearising the plant states about zero slip angle. It should give the same results as the time-invariant filter derived in Section 2.2.
- EKF1: First-order extended Kalman filter. A linearised approximation to the plant states is found at each time step.
- EKF2: Second-order extended Kalman filter. A quadratic approximation to the plant states is found at each time step.
- UKF: Unscented Kalman filter. The nonlinear state covariance functions are approximated at each time step using an unscented transform.
- FSF: Full state feedback: The state estimate $\hat{\mathbf{x}}$ is equal to the real states \mathbf{x} , removing the effects of sensory dynamics.

Detailed derivations of each of the extended Kalman filters are given in [52]. The calculations consist of two stages at each time step: First the *predict* stage in which a new state estimate is predicted from the previous estimate using an internal model of the plant, then the *update* stage where the estimate is updated based on the difference between predicted and measured plant outputs. Unlike the linear Kalman filter, the covariance matrix \mathbf{P} varies with time, and is also predicted and updated in these two steps. The initial covariance matrix \mathbf{P}_0 is calculated by finding the steady-state solution for the plant linearised about zero slip angle from Equation (27), using the Matlab function *kalman*. However, this gives the covariance matrix after the prediction step only, so it must be passed through the update step to ensure compatibility with the extended Kalman filters.

3.3. Controllers

Controllers are designed to calculate an optimal plant input $\hat{\delta}$, using an internal model of the reduced plant described in Section 3.1. True optimal control for a nonlinear plant is computationally intensive, therefore, several simplified controllers are implemented which linearise the system dynamics. These linearised controllers are based on MPC, which involves calculating an optimal control sequence $\hat{\delta}$ up to the prediction horizon T_p , then taking the first of these as the next control input $\hat{\delta}$. For a linear plant this gives the same solution as an LQR controller [3].

Ignoring any white noise inputs, the nonlinear reduced plant equations can be written in discrete-time state-space form:

$$\begin{aligned}\mathbf{x}_R(k+1) &= \mathbf{A}_R(\mathbf{x}_R(k)) + \mathbf{B}_R\hat{\delta}(k), \\ \hat{\mathbf{y}}(k) &= \mathbf{C}_R\mathbf{x}_R(k).\end{aligned}\tag{43}$$

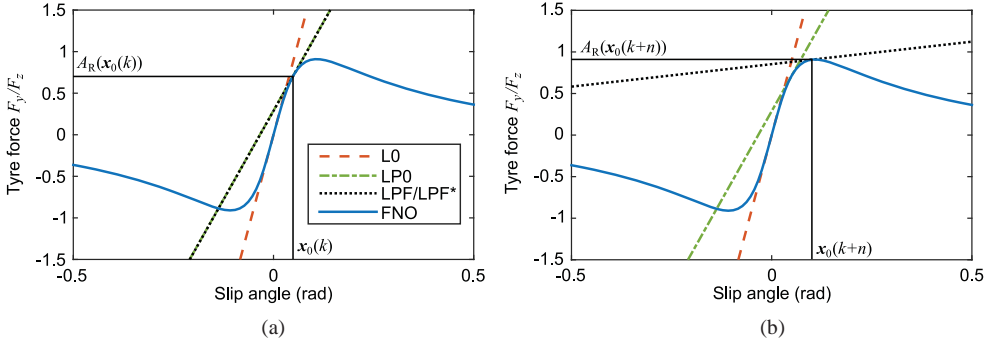


Figure 5. Illustration of linearisation carried out by different controllers. (a) Current time and (b) Future prediction.

The nonlinear function A_R can be linearised about states \mathbf{x}_L with the approximation:

$$A_R(\mathbf{x}_R(k+n)) \approx A_R(\mathbf{x}_L(k+n)) + \hat{\mathbf{A}}_n(\mathbf{x}_R(k+n) - \mathbf{x}_L(k+n)), \quad (44)$$

where $\hat{\mathbf{A}}_n$ is the Jacobian $dA_R/d\mathbf{x}$ evaluated at $\mathbf{x}_L(k+n)$. Five different model predictive controllers are implemented, with varying levels of approximation to the nonlinear plant dynamics as illustrated in Figure 5:

- L0: Linearisation about zero slip angle, $\mathbf{x}_L(k+n) = \mathbf{0}$. This gives the same result as the LQR controller derived in Section 2.3 (assuming the prediction horizon is sufficiently long).
- LP0: Linearisation about the initial state $\mathbf{x}_R(k)$ of the prediction horizon [46,48,50]. $\hat{\mathbf{A}}$ is constant over the prediction horizon, linearised about $\mathbf{x}_L(k+n) = \mathbf{x}_R(k)$.
- LPF: Linearisation about the full predicted trajectory [43,51]. The solution starts from a nominal state trajectory $\mathbf{X}_{R0} = [\mathbf{x}_{R0}(k) \cdots \mathbf{x}_{R0}(k+N_p - N_{vi} + 1)]$, which is the previous optimal sequence shifted by a single time step. The linearised matrix $\hat{\mathbf{A}}_n$ is calculated about each nominal state $\mathbf{x}_L(k+n) = \mathbf{x}_{R0}(k+n)$.
- LPF*: LPF constrained to stop the slip angles exceeding α_{\max} , plus a constraint on the maximum change in $\hat{\delta}$ from the nominal control sequence [44].
- FNO: Full nonlinear optimisation [45,47]. The full nonlinear equations are used to predict the plant trajectory up to the prediction horizon.

3.3.1. Linearised MPC

Four of the controllers simplify the control task by linearising the plant states. They are derived based on the linear MPC method of [3], adapted for a nonlinear plant using methods similar to [51]. For simplicity the control horizon is chosen to equal the prediction horizon. In each case the control sequence $\hat{\delta}$ is replaced with a nominal sequence $\hat{\delta}_0$ plus a small change $\Delta\hat{\delta}$, and similarly for $\hat{\gamma}$. The linearised dynamics then give the approximate

relationship $\Delta\hat{\mathbf{y}} \approx \mathbf{\Theta}\Delta\hat{\boldsymbol{\delta}}$. The aim of the controller is to minimise the cost function:

$$\begin{aligned} J &= \sum_{n=1}^{N_p-N_{vi}+1} \{q_e e(k+n)^2 + q_\delta \hat{\boldsymbol{\delta}}(k+n)^2\} \\ &= \sum_{n=1}^{N_p-N_{vi}+1} \{q_e (\hat{\mathbf{y}}_{\text{ref}}(k+n) - \hat{\mathbf{y}}(k+n))^2 + q_\delta \hat{\boldsymbol{\delta}}(k+n)^2\}. \end{aligned} \quad (45)$$

Note that unlike Equation (29) the cost only applies over the finite prediction horizon. Writing the values of $\hat{\mathbf{y}}_{\text{ref}}$, $\hat{\mathbf{y}}$ and q_δ over the prediction horizon as vectors $\hat{\mathbf{y}}_{\text{ref}}$, $\hat{\mathbf{y}}$ and $\hat{\boldsymbol{\delta}}$, and removing the $\hat{\mathbf{y}}_{\text{ref}}^2$ term which is independent of the control input, Equation (45) can be written as

$$J = q_e \hat{\mathbf{y}}^T \hat{\mathbf{y}} - 2q_e \hat{\mathbf{y}}_{\text{ref}}^T \hat{\mathbf{y}} + q_\delta \hat{\boldsymbol{\delta}}^T \hat{\boldsymbol{\delta}}. \quad (46)$$

Replacing the control sequence $\hat{\boldsymbol{\delta}}$ with a nominal sequence $\hat{\boldsymbol{\delta}}_0$ plus a small change $\Delta\hat{\boldsymbol{\delta}}$, and similarly for $\hat{\mathbf{y}}$, gives

$$J = q_e (\hat{\mathbf{y}}_0 + \Delta\hat{\mathbf{y}})^T (\hat{\mathbf{y}}_0 + \Delta\hat{\mathbf{y}}) - 2q_e \hat{\mathbf{y}}_{\text{ref}}^T (\hat{\mathbf{y}}_0 + \Delta\hat{\mathbf{y}}) + q_\delta (\hat{\boldsymbol{\delta}}_0 + \Delta\hat{\boldsymbol{\delta}})^T (\hat{\boldsymbol{\delta}}_0 + \Delta\hat{\boldsymbol{\delta}}) \quad (47)$$

Multiplying out and removing any independent terms:

$$J = q_e \Delta\hat{\mathbf{y}}^T \Delta\hat{\mathbf{y}} + 2q_e (\hat{\mathbf{y}}_0 - \hat{\mathbf{y}}_{\text{ref}})^T \Delta\hat{\mathbf{y}} + q_\delta \Delta\hat{\boldsymbol{\delta}}^T \Delta\hat{\boldsymbol{\delta}} + 2q_\delta \hat{\boldsymbol{\delta}}_0^T \Delta\hat{\boldsymbol{\delta}} \quad (48)$$

and with the linearised dynamics $\Delta\hat{\mathbf{y}} = \mathbf{\Theta}\Delta\hat{\boldsymbol{\delta}}$:

$$J = \Delta\hat{\boldsymbol{\delta}}^T (q_e \mathbf{\Theta}^T \mathbf{\Theta} + q_\delta \mathbf{I}) \Delta\hat{\boldsymbol{\delta}} + 2(q_e (\hat{\mathbf{y}}_0 - \hat{\mathbf{y}}_{\text{ref}})^T \mathbf{\Theta} + q_\delta \hat{\boldsymbol{\delta}}_0^T) \Delta\hat{\boldsymbol{\delta}}, \quad (49)$$

which is in the form of a quadratic program. If there are no constraints this can be solved using QR decomposition as in [3], however with constraints it can be solved using a solver such as QPC [55] or Matlab's *quadprog* function.

Similarly to the linear MPC derivation in [3], however with linearised matrices $\hat{\mathbf{A}}_n$ predicted n time steps ahead of the current time step k , $\mathbf{\Theta}$ is found from:

$$\mathbf{\Theta}(k) = \begin{bmatrix} \mathbf{C}_R \mathbf{B}_R & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_R \hat{\mathbf{A}}_1 \mathbf{B}_R & \mathbf{C}_R \mathbf{B}_R & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_R \hat{\mathbf{A}}_2 \hat{\mathbf{A}}_1 \mathbf{B}_R & \mathbf{C}_R \hat{\mathbf{A}}_2 \mathbf{B}_R & \mathbf{C}_R \mathbf{B}_R & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_R \left(\prod_{i=1}^{N_p-N_{vi}} \hat{\mathbf{A}}_i \right) \mathbf{B}_R & \mathbf{C}_R \left(\prod_{i=2}^{N_p-N_{vi}} \hat{\mathbf{A}}_i \right) \mathbf{B}_R & \mathbf{C}_R \left(\prod_{i=3}^{N_p-N_{vi}} \hat{\mathbf{A}}_i \right) \mathbf{B}_R & \cdots & \mathbf{C}_R \mathbf{B}_R \end{bmatrix} \quad (50)$$

For LPF and LPF*, the nominal control sequence $\hat{\boldsymbol{\delta}}_0$ is the previous optimal control sequence shifted by a single time step. The nonlinear plant equations are then evaluated over the prediction horizon to get $\hat{\mathbf{y}}_0$. For L0 and LP0, the nominal control sequence $\hat{\boldsymbol{\delta}}_0$ is

zero. However, the non-zero initial state $\mathbf{x}_R(k)$ leads to non-zero $\hat{\mathbf{y}}_0$. For L0, as in [3]:

$$\hat{\mathbf{y}}_0 \approx \Psi \mathbf{x}_R(k), \quad \Psi = [\mathbf{C}_R \hat{\mathbf{A}} \quad \mathbf{C}_R \hat{\mathbf{A}}^2 \quad \dots \quad \mathbf{C}_R \hat{\mathbf{A}}^{N_p - N_{vi}}]^T. \quad (51)$$

For LP0:

$$\begin{aligned} \mathbf{x}_R(k+1) &= A_R(\mathbf{x}_R(k)) \\ \mathbf{x}_R(k+2) &= A_R(\mathbf{x}_R(k+1)) \approx A_R(\mathbf{x}_R(k)) + \hat{\mathbf{A}}(\mathbf{x}_R(k+1) - \mathbf{x}_R(k)) \\ &= (\mathbf{I} + \hat{\mathbf{A}})A_R(\mathbf{x}_R(k)) - \hat{\mathbf{A}}\mathbf{x}_R(k). \end{aligned} \quad (52)$$

Over the whole prediction horizon this leads to

$$\hat{\mathbf{y}}_0 \approx \Psi_x \mathbf{x}_R(k) + \Psi_A A_R(\mathbf{x}_R(k)), \quad (53)$$

where

$$\Psi_x = \begin{bmatrix} \mathbf{0} \\ -\mathbf{C}_R \hat{\mathbf{A}} \\ -\mathbf{C}_R (\hat{\mathbf{A}}^2 + \hat{\mathbf{A}}) \\ \vdots \\ -\mathbf{C}_R \sum_{i=1}^{N_p - N_{vi}} (\hat{\mathbf{A}}^i) \end{bmatrix}, \quad \Psi_A = \begin{bmatrix} \mathbf{C}_R \\ \mathbf{C}_R (\hat{\mathbf{A}} + \mathbf{I}) \\ \mathbf{C}_R (\hat{\mathbf{A}}^2 + \hat{\mathbf{A}} + \mathbf{I}) \\ \vdots \\ \mathbf{C}_R \sum_{i=0}^{N_p - N_{vi}} (\hat{\mathbf{A}}^i) \end{bmatrix}. \quad (54)$$

3.3.2. LPF*

Controller LPF* is the same as LPE, however, it includes constraints to limit the solution space to values more likely to give a successful outcome. Firstly, the linearisation assumes that the changes in optimal control strategy $\Delta \hat{\delta}$ are small. Therefore, constraints are added to limit these changes, in the form:

$$\begin{aligned} \Delta \hat{\delta} &< \Delta \hat{\delta}_{\max}, \\ -\Delta \hat{\delta} &< \Delta \hat{\delta}_{\max}. \end{aligned} \quad (55)$$

The value of $\Delta \hat{\delta}_{\max}$ must be small enough to stop the optimal solution from moving too far from the nominal solution; however, large enough that it does not limit the speed with which the controller can respond to disturbances. Simulations were run with different values of $\Delta \hat{\delta}_{\max}$, and a value of 0.1 rad* was found to be suitable (the superscript * after a unit indicates that the value applies to a filtered version of that unit).

Another potential issue with linearised controllers is that they can become unstable if the linearised force/slip characteristic becomes negative. Therefore, additional constraints are added to limit the slip angles α_f and α_r to less than α_{\max} , where α_{\max} is defined to be the maximum of the force-slip curve. Using the vehicle defined in Section 5, the slip angles

are found from linear functions of the first 3 plant states:

$$\alpha_f = -v/U - l_f\omega/U + \delta/G, \quad \alpha_r = -v/U + l_r\omega/U, \quad (56)$$

therefore the equations for α_f and α_r are linearised as

$$\alpha_f = \alpha_{f0} + \Theta_{\alpha f} \Delta \hat{\delta}, \quad \alpha_r = \alpha_{r0} + \Theta_{\alpha r} \Delta \hat{\delta}, \quad (57)$$

$\Theta_{\alpha f}$ and $\Theta_{\alpha r}$ are calculated using Equation (50), replacing C_R with $C_{\alpha f}$ or $C_{\alpha r}$, where

$$C_{\alpha f} = \begin{bmatrix} -1/U & -l_f/U & 1/G & 0 \end{bmatrix}, \quad C_{\alpha r} = \begin{bmatrix} -1/U & l_r/U & 0 & 0 \end{bmatrix}. \quad (58)$$

The constraint equations therefore become

$$\begin{aligned} \Theta_{\alpha f} \Delta \hat{\delta} &< \alpha_{\max} - \alpha_{f0}, & \Theta_{\alpha r} \Delta \hat{\delta} &< \alpha_{\max} - \alpha_{r0}, \\ -\Theta_{\alpha f} \Delta \hat{\delta} &< \alpha_{\max} + \alpha_{f0}, & -\Theta_{\alpha r} \Delta \hat{\delta} &< \alpha_{\max} + \alpha_{r0}. \end{aligned} \quad (59)$$

3.3.3. Full nonlinear optimisation

In order to find a solution which considers the nonlinear dynamics of the plant in full without any approximation, a nonlinear optimiser can be used. Matlab's *fminunc* function is a versatile nonlinear optimiser, although it is much more computationally expensive than the linearised alternatives. As with the linearised MPC, the optimisation begins from a nominal steering command sequence $\hat{\delta}_0$, which is the previous optimal steering sequence shifted by a single time step. The optimiser then iteratively evaluates the cost function for different values of $\Delta \hat{\delta}$, in each case calculating the full nonlinear reduced plant equations over the prediction horizon. Gradient-based optimisers such as *fminunc* use the Jacobian of the cost function in order to estimate the direction of the optimal solution. Although this can be estimated by the optimiser, it is much faster to calculate the Jacobian explicitly. This can be achieved by linearising the dynamics similarly to the LPF controller. The derivative of the cost function J about $\hat{\delta}$ is equal to the linear term in Equation (49):

$$\frac{dJ}{d\hat{\delta}} = 2(q_e(\hat{y} - \hat{y}_{\text{ref}})^T \Theta + q_\delta \hat{\delta}^T). \quad (60)$$

4. Model parameter values

The performance of the driver model depends on the values of various transfer functions and parameters. Some of these are properties of the driving conditions, for example, the vehicle, motion filters and forcing function filters and amplitudes. In general it is assumed that the driver's internal model matches the true values of these exactly. Other values relate to physical properties of the human driver; therefore, work has been carried out to find suitable values for these parameters. A review of the literature was undertaken to choose appropriate physiological models for the driver's sensory and neuromuscular dynamics [4]. An approximation to drivers' neuromuscular dynamics can be made using a second-order

filter [56,57]:

$$H_{nm}(s) = \frac{\omega_{nm}^2}{s^2 + 2\zeta_{nm}\omega_{nm}s + \omega_{nm}^2}. \quad (61)$$

Transfer functions for the SCCs and otoliths are [58]:

$$H_{SCC}(s) = \frac{458.4s^2}{(80s + 1)(5.73s + 1)}, \quad (62)$$

$$H_{oto}(s) = \frac{0.4(10s + 1)}{(5s + 1)(0.016s + 1)}. \quad (63)$$

Motivated by previous work in the aerospace industry [37–39], experiments were carried out by human drivers in a driving simulator, and an identification procedure was used to find suitable values of the other parameters in order to fit the model as closely as possible to the measured data [40,42]. In [40], the process noise amplitude W was found to be proportional to the RMS steering angle δ , giving the relationship:

$$W = \text{RMS}(\delta)/\text{SNR}_W, \quad (64)$$

where SNR_W is the identified signal-to-noise ratio. Note that the SNR is expressed as the ratio of the RMS values rather than the mean-square ratio more commonly used in communications engineering. Similarly, in [42] the measurement noise amplitudes were found to be proportional to the equivalent RMS signals, however, with thresholds η below which the measurement noise remains constant. For example:

$$V_a = \begin{cases} \text{RMS}(a_{ve})/\text{SNR}_a & \text{RMS}(a_{ve}) > \eta_a, \\ \eta_a/\text{SNR}_a & \text{RMS}(a_{ve}) < \eta_a. \end{cases} \quad (65)$$

This is consistent with sensory noise and threshold characteristics measured in previous studies [27–33]. Measurement noise of variance V_ϕ^2 is added to each previewed visual angle ϕ_{vi} ; however, the influence of these measurements would vary with prediction horizon N_p if V_ϕ was kept constant. Therefore, the combined standard deviation σ_ϕ is calculated as for V_a in Equation (65), and V_ϕ is found from $V_\phi = \sigma_\phi \sqrt{N_p} + \bar{1}$.

A single set of parameter values was identified in [42] to fit the results of 51 trials carried out over a range of conditions. The identified values are given in Table 1. The only parameters which are not given are the cost function weights, because these parameters are a choice of the driver rather than a representation of human physiology. They depend on both the driving conditions and the individual effort made by the driver. The controller is affected only by the relative values of the weightings therefore q_e is set to 1 m^{-2} . In the experiments, values of q_δ between around 0.1 and 2 rad^{-2*} were identified.

Table 1. Driver model parameter values identified in [42].

SNR_a	SNR_ω	SNR_ϕ	SNR_e	SNR_W	τ_{vi} (s)	τ_{ve} (s)
–	–	–	–	–	–	–
0.390	0.406	1.46	0.901	2.28	0.19	0.23
T_p (s)	ω_{nm} (rad/s)	ζ_{nm} (–)	η_a (m/s^{2*})	η_ω (rad/s^*)	η_ϕ (rad)	η_e (m)
0.87	14.3	0.537	0.221	0.0235	0.0129	0.0559

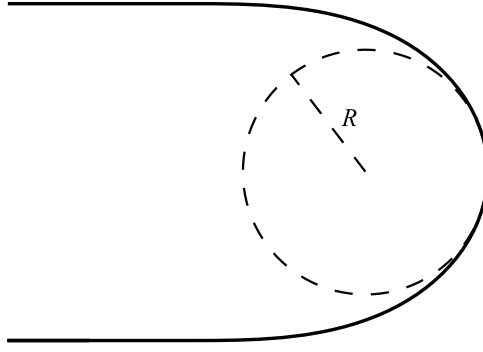


Figure 6. Target path used in simulations.

5. Simulations

Simulations can be carried out to compare the various state estimators and controllers derived in Section 3, in order to evaluate which may be suitable for modelling driver steering control. Each simulation involves driving round a 180° corner with a clothoid curvature profile, as shown in Figure 6. In order to test the controllers at different operating points on the force-slip curve, the minimum radius R of the corner is varied.

The vehicle is based on the two degree-of-freedom single-track model, shown in Figure 7. Using Newton's second law to balance lateral forces and moments gives

$$\dot{v} = \frac{F_{yf}}{m} + \frac{F_{yr}}{m} - U\omega, \quad \dot{\omega} = \frac{l_f F_{yf}}{I} - \frac{l_r F_{yr}}{I}. \quad (66)$$

With linear tyres the cornering stiffness is constant, so:

$$F_{yf} = C_f \alpha_f, \quad F_{yr} = C_r \alpha_r, \quad (67)$$

where α_f and α_r are given by Equation (56). These equations can be written in continuous state-space form, adding disturbances f_v and f_ω to the lateral and angular accelerations \dot{v} and $\dot{\omega}$:

$$\begin{Bmatrix} \frac{dv}{dt} \\ \frac{d\omega}{dt} \end{Bmatrix} = \begin{bmatrix} \frac{-(C_f + C_r)}{Um} & \frac{-(l_f C_f - l_r C_r)}{Um} - U \\ \frac{-(l_f C_f - l_r C_r)}{UI} & \frac{-(l_f^2 C_f + l_r^2 C_r)}{UI} \end{bmatrix} \begin{Bmatrix} v \\ \omega \end{Bmatrix} + \begin{bmatrix} \frac{C_f}{mG} & 1 & 0 \\ \frac{l_f C_f}{IG} & 0 & 1 \end{bmatrix} \begin{Bmatrix} \delta \\ f_v \\ f_\omega \end{Bmatrix},$$

$$\dot{\mathbf{x}}_V(t) = \mathbf{A}_{Vc} \mathbf{x}_V(t) + \mathbf{B}_{Vc} \begin{Bmatrix} \delta(t) & f_v(t) & f_\omega(t) \end{Bmatrix}^T. \quad (68)$$

The continuous matrices \mathbf{A}_{Vc} and \mathbf{B}_{Vc} are then discretised to give \mathbf{A}_V and \mathbf{B}_V . The outputs v and ω are equal to the states, therefore, $\mathbf{C}_V = \mathbf{I}$ and $\mathbf{D}_V = \mathbf{0}$.

For the nonlinear vehicle, the lateral tyre forces are given by the 'magic formula' [59]:

$$F_y = F_z D \sin(C \tan^{-1}(B(1 - E)\alpha + E \tan^{-1}(B\alpha))). \quad (69)$$

For small slip angles this function is approximately linear, so an equivalent linear vehicle can be found by letting $C_f = dF_{yf}/d\alpha_f = BCDF_{zf}$ and $C_r = dF_{yr}/d\alpha_r = BCDF_{zr}$, and

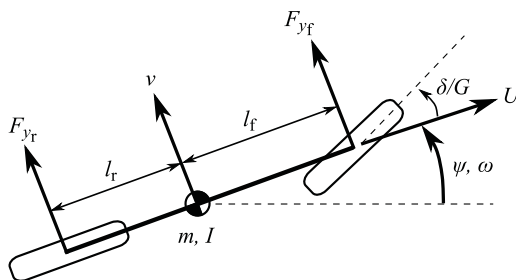


Figure 7. Single-track vehicle model used in simulations.

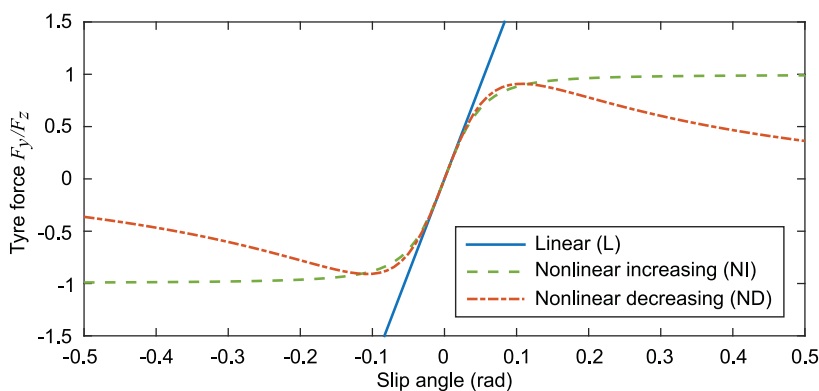


Figure 8. Characteristics of the three tyres used in the simulations.

Table 2. Nonlinear tyre parameters.

Tyre	B	C	D	E
Nonlinear increasing (NI)	12	1.5	1	1
Nonlinear decreasing (ND)	9	2.2	0.909	0.5

Table 3. Vehicle model parameters.

Vehicle	m (kg)	l_f (m)	l_r (m)	I (kgm ²)	G (-)	U (m/s)	F_{zf} (N)	F_{zr} (N)
Understeering	650	1.85	1.65	450	10	40	9500	15,000
Oversteering	650	1.85	1.65	450	10	40	12,000	13,000

this can be used to initialise the simulation. Three different tyres are simulated: a linear tyre (L), a nonlinear tyre with force monotonically increasing as a function of slip angle (NI), and a nonlinear tyre with force decreasing past the friction limit (ND). The force-slip characteristics of these three tyres are shown in Figure 8, and the nonlinear tyre parameters are given in Table 2. All three tyres have the same cornering stiffness at zero slip angle ($C_f/F_{zf} = C_r/F_{zr} = 18 \text{ rad}^{-1}$). Two different vehicles are simulated, one with understeering characteristics and one with oversteering characteristics. This is achieved by varying the balance of the vertical loads F_{zf} and F_{zr} between the front and rear tyres. The parameters for the two vehicles are given in Table 3.

Simulations are run with each combination of state estimator and controller derived in Section 3, controlling both the oversteering and understeering vehicles with the three different tyres. Initially, simulations are run with no disturbances and without adding driver noise. A second set of simulations are then run with filtered white noise disturbances, with amplitudes $W_v = 5 \text{ m/s}^2$ and $W_\omega = 5 \text{ rad/s}^2$ and spectra:

$$H_{fv}(s) = H_{f\omega}(s) = \left(\frac{s}{s+1} \right) \left(\frac{10}{s+10} \right). \quad (70)$$

Further simulations are also carried out with simulated driver noise. The motion scaling filters $H_{ma}(s)$ and $H_{m\omega}(s)$ are always set to 1, to represent a real driving scenario.

Parameter values for the model are taken from the experimental conditions and the values given in Section 4. The steering cost weight q_δ is set to 0.5, and the prediction horizon is extended to 2 s since T_p also represents the control horizon for the nonlinear model. As explained in Section 4, the process and measurement noise amplitudes depend on the RMS values of the equivalent signals. However, these RMS values are not known until after the simulation has been run, so an iterative procedure would be needed to find the RMS signal values for each condition. To save time, fixed estimates of the RMS signal values are used to calculate the noise amplitudes, equal to 0.1 for $\text{RMS}(\omega_{ve})$, $\text{RMS}(e_{vi})$, $\text{RMS}(\phi_{vi})$ and $\text{RMS}(\delta)$, and 1 for $\text{RMS}(a_{ve})$. The target signal f_k is not filtered white noise as assumed by the model, but follows a transient corner profile. For the simulations the internal model spectrum of the target is set to white noise ($H_{fk}(s) = 1$), however, further work is needed to understand fully how drivers react to transient target and disturbance functions.

6. Results

Two main outcomes of the simulations can be analysed to compare the different controllers and state estimators: the time each simulation takes to run and how well each combination is able to minimise the cost function.

6.1. Simulation time

The time taken to run each simulation is an important factor when considering which methods could be used practically in engineering applications. It can also be used as a measure of the mental load involved, and hence the likelihood of a human driver being able to use each method. The human brain contains bottlenecks which limit the capacity of drivers to carry out simultaneous tasks [60]; therefore, a very intensive online calculation could use up too many mental resources. Instead, drivers may use learned responses or simplifications similar to the linearised controllers and state estimators derived in Section 3.

The average time taken to run one simulation with each combination of controller and state estimator is shown in Figure 9. There is a difference of several orders of magnitude between the fastest and slowest combinations, and the simulation time increases with the accuracy of the approximation to the true nonlinear dynamics. This could reflect the trade-off the driver must make between accuracy and mental load. In particular, the FNO controller is around 20 times slower than the next slowest controller, highlighting the complexity of carrying out true optimal control for a nonlinear plant.

Controllers	L0	0.33	20	29	30	100
	LP0	27	44	52	53	130
	LPF	38	52	60	60	130
	LPF*	60	74	82	83	160
	FNO	1200	1100	1100	1100	1200
		FSF	LKF	EKF1	EKF2	UKF
		State Estimators				

Figure 9. Average time taken (s) for simulation to run using each combination of controller and state estimator.

6.2. Controller performance

The performance of each combination of state estimator and controller can be evaluated by finding the total value of the controller cost function J given in Equation (29) over each simulation. This can be used to investigate how close each combination is to the optimal solution; however, further experimentation is required to determine the extent to which real drivers are able to carry out optimal control for a nonlinear vehicle.

For the linear tyre, the controllers all perform similarly, as expected. A comparison of the performance of the different controllers for the nonlinear tyres, with no disturbances or driver noise, is shown in Figure 10. The cost has been normalised by the cost found using the FNO controller, hence the FNO controller always has a normalised cost of 1. Figure 10 shows that FNO is generally the best performing controller, although for radii of 42 and 50 m with the oversteering ND vehicle other controllers perform slightly better. This is likely to be a result of the finite prediction horizon, which may not be sufficiently long to give the optimal control response under all conditions. LPF* performs similarly to FNO in most conditions, except for the oversteering ND vehicle at lower radii. LPF also performs as well for the oversteering NI vehicle, showing that the constraints are not necessary in this condition. LP0 performs better than L0, and sometimes better than LPF. The performances of all controllers converge at larger radii, as the vehicle response is close to linear for small slip angles.

The performance of the different controllers with added disturbances and/or driver noise is similar to the results shown in Figure 10, although with larger overall costs. For the oversteering vehicle with ND tyres LPF* becomes unstable at a higher radius than seen in Figure 10, and does not regain stability at the lowest radii. This may be because the disturbances push the vehicle past the friction peak of the tyres, outside the constraints of the LPF* controller. Controller FNO also becomes unstable in some cases, which may be a result of the short prediction horizon or the fact that the FNO controller is not guaranteed to be robust to disturbances for a nonlinear plant.

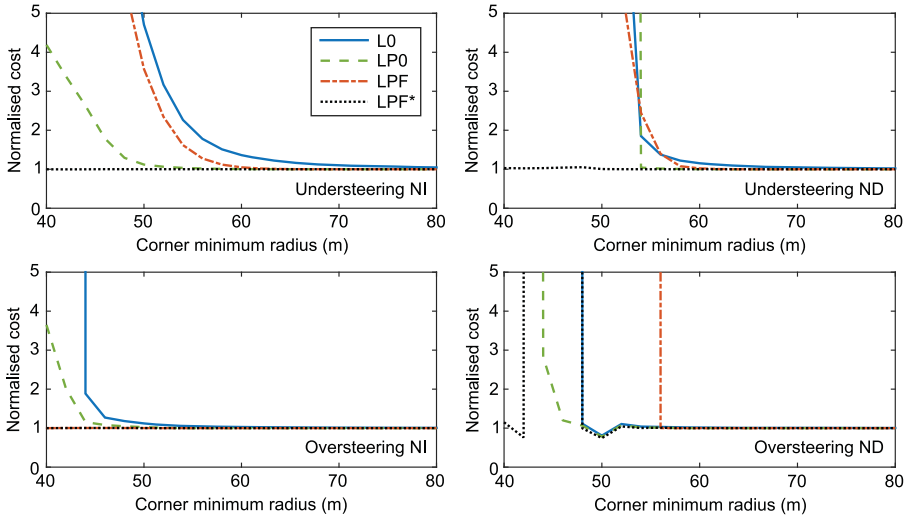


Figure 10. Comparison of the performance of each controller for simulations without added disturbances or noise and with FSF. The total value of the cost function over each simulation has been normalised by the value found for the FNO controller.

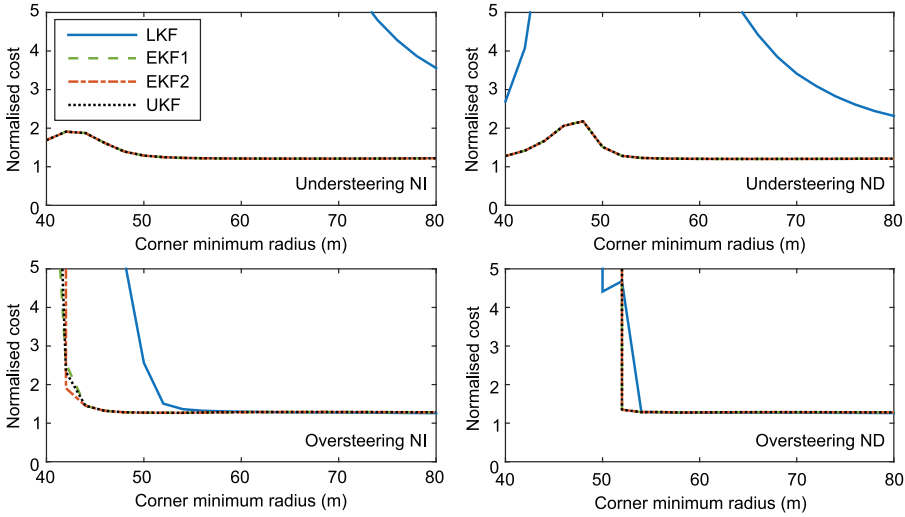


Figure 11. Comparison of the performance of each state estimator for simulations without added disturbances or noise and with the FNO controller. The total value of the cost function over each simulation has been normalised by the value found for FSF.

6.3. State estimator performance

The performance of the different state estimators with no disturbances or noise is compared in Figure 11. The costs have all been normalised by the cost found for FSF. The linear Kalman filter performs worst as expected, particularly for the understeering vehicle. All of the extended Kalman filters perform very similarly; however, they always result in a cost at least 20% higher than FSF. This may be a result of the target signal which is not formed from

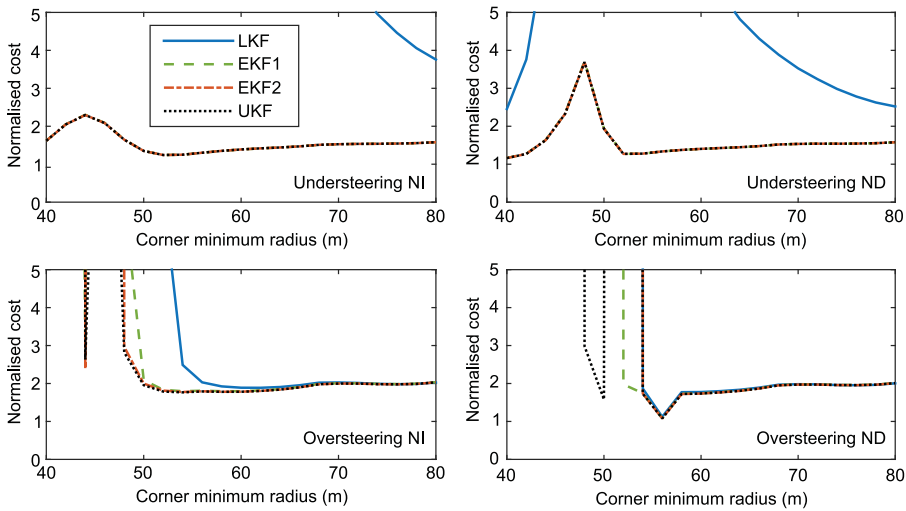


Figure 12. Comparison of the performance of each state estimator for simulations with added disturbances and driver noise and with the FNO controller. The total value of the cost function over each simulation has been normalised by the value found for FSF.

filtered white noise, causing sub-optimal state estimates from the extended Kalman filters. The significance of this effect can be altered by changing the relative noise amplitudes in the model. For the oversteering vehicle, the extended Kalman filters are unable to control the vehicle at small radii, particularly with ND tyres. For the linear vehicle, all state estimators perform well in all conditions; however, they still perform slightly worse than FSF.

The performance of the state estimators with added disturbances and driver noise is shown in Figure 12. The difference between the extended Kalman filters and FSF is larger than for the results shown in Figure 11. This is mainly a result of the driver's sensory delays, which stop the disturbances being perceived until around 0.2 s after they have occurred. In contrast, with FSF the driver has access to all the delay states so is aware of the disturbance straight away, and can react quicker in order to achieve a lower cost. Compared with Figure 11, the extended Kalman filters become unstable for the oversteering vehicle at a larger radius. Some differences between state estimators are seen in this region; however, generally the extended Kalman filters perform similarly in stable conditions. Similar results are found with disturbances and driver noise separately, although differences between FSF and the state estimators are smaller than in the combined case.

6.4. Optimal performance

The costs presented in Sections 6.2 and 6.3 are normalised by the results for the best-performing controller (FNO) and state estimator (FSF) to highlight differences between the various implementations. To demonstrate the optimal performance achieved in the simulations, the total absolute costs for a combination of FNO and FSF are plotted in Figure 13. For the understeering vehicle, the total cost increases significantly as the corner radius decreases, since the driver is required to apply a larger steering angle to account for the vehicle understeer. For the oversteering vehicle, the total cost generally does not change

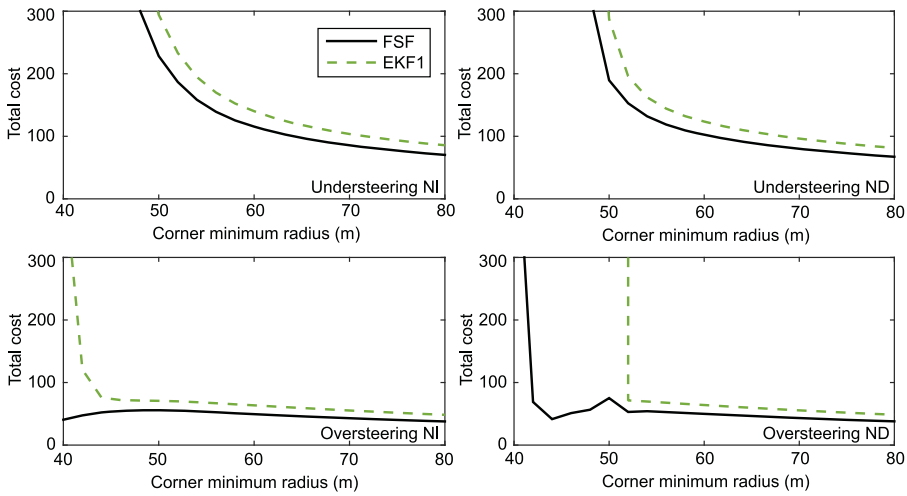


Figure 13. Total value of the cost function J for simulations with no added disturbances or driver noise and with the FNO controller. The best possible state estimator (FSF) is compared with the best-performing realistic state estimator (EKF1).

significantly with corner radius. However with ND tyres large costs are seen for the smallest radii, which indicates that the controller becomes unstable. This may be due to the limited prediction and control horizons, resulting in a sub-optimal control performance.

Although the best performance is achieved with FSF, in reality drivers must estimate the plant states based on noisy sensory measurements. All extended Kalman filters were found to perform similarly in Section 6.3; therefore, EKF1 can be chosen to represent the optimal performance of a real driver. The total cost for a combination of FNO and EKF1 is also plotted in Figure 13. For the understeering vehicle, the performance is similar to FSF, just with a slightly higher cost. For an oversteering vehicle, the costs at low radii are very large, due to the driver–vehicle system becoming unstable. This happens at a larger radius with the ND tyres than the NI tyres. Results with added disturbances and driver noise are similar to those shown in Figure 13, although with higher costs.

7. Discussion

The results of the simulations presented in Section 6 can give some insight into the influence of a driver's sensory dynamics on the control of a vehicle. They can also be used to evaluate the controllers and state estimators for use in a driver model, both in terms of how well they model a realistic driver and the practicality of using them in engineering applications. It is possible to suggest improvements to the controllers which may allow them to run faster or more accurately, or better represent the human driver.

7.1. Effect of sensory dynamics

The effect of a driver's sensory dynamics on the control of a nonlinear vehicle can be investigated by comparing the performance of the extended Kalman filters with the results using FSF in Figures 11–13. In all cases the simulated driver's performance is worse with

a state estimator than with FSF, showing that human drivers' sensory limitations reduce their control performance. For small-radius corners with an understeering vehicle the driver-vehicle system becomes unstable with a realistic state estimator, which indicates that sensory dynamics play a role in the robustness of the driver-vehicle system. This is an important result for the design of safe, stable vehicles or stability control systems. The difference between states estimators and FSF is larger when there are disturbances added to the vehicle, showing that sensory dynamics are more important for disturbance-rejection than target-following. This is intuitively reasonable, as drivers are able to preview the upcoming target in order to compensate for sensory delays. Adding simulated driver noise gives similar results to adding disturbances to the vehicle, showing that driving will always involve a certain amount of feedback control even in the absence of external disturbances. Even with no disturbances or noise FSF still performs better than the state estimators, which may be a result of using a target which is not formed from filtered white noise. Further work is necessary to understand how drivers react to transient target or disturbance signals.

7.2. *Controllers and state estimators*

In general there is an inverse relationship between the performance and the computational speed of each controller. FNO is the best performing controller as expected, however LPF* performs as well in many conditions and is much faster. The driver model has particular difficulty controlling an oversteering vehicle with ND tyres, and even FNO sometimes becomes unstable in these conditions. This may be in part due to the finite control horizon, however increasing this would increase the time taken to simulate the controller. Also, while the FNO controller should give an optimum performance without disturbances (with a sufficiently long control horizon), it is not necessarily a robust controller when disturbances are added. Linearised controllers are much faster, however, they have difficulty operating past the friction limit for ND tyres, so need to be constrained to operate below this limit.

Differences between the state estimators are smaller in general than for the controllers. LKF performs worse than the other state estimators for nonlinear tyres, as expected. FSF performs best, however, this is not a good representation of a real driver who is affected by sensory dynamics and delays. In most cases, all the nonlinear Kalman filters perform similarly, so it is sensible to choose EKF1 which takes the least time to simulate.

While the results presented in Section 6 indicate which combinations of controller and state estimator are most effective in minimising the cost function, they are not necessarily representative of a real driver. For a linear vehicle drivers have been shown to steer close to the optimal solution with full or scaled motion; however, they can have difficulty learning accurate internal models of complicated motion filters [40,42]. Drivers may have similar difficulties with nonlinear vehicle dynamics, and might make simplifications similar to the linearised controllers or state estimators in order to reduce their mental load. It has been suggested that inexperienced drivers may hold a limited number of linearised internal models of the nonlinear plant [51]. Alternatively, drivers may use learned responses to achieve close to optimal control without a large 'online' mental load. Experiments are planned using a driving simulator to test how the performance of real drivers compares to the simulations.

7.3. Improvements to controllers

Various methods can be used to improve the optimality and efficiency of the controllers. It has already been shown that the performance of controller LPF can be improved by adding constraints to limit slip angles and keep the solution close to the linearised operating point (LPF*). Similar constraints were tested for LP0; however, they were found not to be effective. An advantage of FNO over LPF* is that it is able to operate beyond the limit of tyre friction; however, it may be possible to develop a combined controller that uses LPF* for smaller slip angles and FNO for larger slip angles. There are clearly some issues with robustness, so it would be useful to develop a nonlinear controller that is more robust to disturbances under all conditions.

The practicality of some of the more complex controllers is somewhat limited by the computation time, with each simulation using FNO taking around 20 min to complete. There are various approximations which can be made to try and reduce this time, and if done carefully they may not significantly impact the performance of the controller. The simulations were run at a sample frequency of 100 Hz; however, driver steering control is unlikely to act over such a high bandwidth. It may therefore be possible to reduce the sample frequency, or maintain a high sampling rate for simulating the plant dynamics while running the controller less frequently. Recent research has investigated intermittent control, where the driver updates their control sequence less frequently and relies on their previous computation in the interval between calculations [60]. It may also be possible to calculate some portion of the control strategy offline to reduce the online computational load, which could model drivers' learned behaviours from previous driving experience. Further speed increases may be achieved by writing the algorithms in a compiled language such as C++.

8. Conclusions

A new model of driver steering control incorporating human sensory dynamics has been developed, and the model has been extended to represent the control of a nonlinear vehicle. Simulations were run to compare various controllers and state estimators. In choosing the controller there is generally a trade-off between computation time (which could represent mental load) and controller performance, although a linearised controller was implemented which performs as well as a full nonlinear optimisation in many conditions. All versions of the extended Kalman filter perform similarly; therefore, it is sensible to use the quickest of these which involves a linear approximation. Sensory dynamics affect the control performance and stability of the driver-vehicle system, particularly with external disturbances or driver noise. Experiments are planned to compare the simulations to measured behaviour from human drivers.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the UK Engineering and Physical Sciences Research Council (EP/P505445/1, studentship for Nash).

ORCID

C.J. Nash  <http://orcid.org/0000-0002-7981-1265>

D.J. Cole  <http://orcid.org/0000-0003-3162-701X>

References

- [1] Plöchl M, Edelmann J. Driver models in automobile dynamics application. *Veh Syst Dyn*. 2007;45:699–741.
- [2] Sharp RS, Valtetsiotis V. Optimal preview car steering control. *Veh Syst Dyn Suppl*. 2001;35:101–117.
- [3] Cole DJ, Pick AJ, Odhams AMC. Predictive and linear quadratic methods for potential application to modelling driver steering control. *Veh Syst Dyn*. 2006;44:259–284.
- [4] Nash CJ, Cole DJ, Bigler RS. A review of human sensory dynamics for application to models of driver steering and speed control. *Biol Cybern*. 2016;110:91–116.
- [5] Wolpert DM, Ghahramani Z. Computational principles of movement neuroscience. *Nat Neurosci Suppl*. 2000;3:1212–1217.
- [6] Wolpert DM, Diedrichsen J, Flanagan JR. Principles of sensorimotor learning. *Nat Rev Neurosci*. 2011;12:739–751.
- [7] Ernst MO, Banks MS. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*. 2002;415:429–433.
- [8] Butler JS, Smith ST, Campos JL, et al. Bayesian integration of visual and vestibular signals for heading. *J Vision*. 2010;10:1–13.
- [9] Seilheimer RL, Rosenberg A, Angelaki DE. Models and processes of multisensory cue combination. *Curr Opin Neurobiol*. 2014;25:38–46.
- [10] Grewal MS, Andrews AP. Kalman filtering: theory and practice using MATLAB. 2nd ed. New York: Wiley; 2001.
- [11] Gibson JJ. The perception of the visual world. Boston, MA: Houghton Mifflin; 1950.
- [12] Gordon DA. Static and dynamic visual fields in human space perception. *J Opt Soc Amer*. 1965;372:1296–1303.
- [13] Lappe M, Bremmer F, Van Den Berg AV. Perception of self-motion from visual flow. *Trends Cogn Sci*. 1999;3:329–336.
- [14] Young LR, Meiry JL. A revised dynamic otolith model. *Aerosp Med*. 1968;39:606–608.
- [15] Fernandez C, Goldberg JM. Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey. Parts I to III. *J Neurophysiol*. 1971;34:661–675.
- [16] Fernandez C, Goldberg JM. Physiology of peripheral neurons innervating otolith organs of the squirrel monkey. Parts I to III. *J Neurophysiol*. 1976;39:970–984.
- [17] Hosman RJAW. Pilot's perception and control of aircraft motions [PhD Thesis]. Delft University of Technology; 1996.
- [18] Collins DF, Refshauge KM, Todd G, et al. Cutaneous receptors contribute to kinesthesia at the index finger, elbow, and knee. *J Neurophysiol*. 2005;94:1699–1706.
- [19] Vaitl D, Mittelstaedt H, Saborowski R, et al. Shifts in blood volume alter the perception of posture: Further evidence for somatic graviception. *Int J Psychophysiol*. 2002;44:1–11.
- [20] Poppele RE, Bowman RJ. Quantitative description of linear behavior of mammalian muscle spindles. *J Neurophysiol*. 1970;33:59–72.
- [21] Houck JA, Simon W. Responses of Golgi tendon organs to forces applied to muscle tendon. *J Neurophysiol*. 1967;30:1466–1481.
- [22] Kawakami O, Kaneoke Y, Maruyama K, et al. Visual detection of motion speed in humans : spatiotemporal analysis by fMRI and MEG. *Hum Brain Mapp*. 2002;118:104–118.
- [23] Lam K, Kaneoke Y, Gunji A, et al. Magnetic response of human extrastriate cortex in the detection of coherent and incoherent motion. *Neuroscience*. 2000;97:1–10.
- [24] Aw ST, Todd MJ, Halmagyi GM. Latency and initiation of the human vestibuloocular reflex to pulsed galvanic stimulation. *J Neurophysiol*. 2006;96:925–930.

- [25] Trojaborg W, Sindrup EH. Motor and sensory conduction in different segments of the radial nerve in normal subjects. *J Neurol, Neurosurgery, Psychiatry*. 1969;32:354–359.
- [26] Barnett-Cowan M. Vestibular perception is slow: a review. *Multisensory Res*. 2013;26:387–403.
- [27] Benson AJ, Spencer MB, Stott JR. Thresholds for the detection of the direction of whole-body, linear movement in the horizontal plane. *Aviation Space Environ Med*. 1986;57:1088–1096.
- [28] Benson AJ, Hutt EC, Brown SF. Thresholds for the perception of whole body angular movement about a vertical axis. *Aviation Space Environ Med*. 1989;60:205–213.
- [29] Grabherr L, Nicoucar K, Mast FW, et al. Vestibular thresholds for yaw rotation about an earth-vertical axis as a function of frequency. *Exp Brain Res*. 2008;186:677–681.
- [30] Mallery RM, Olomu OU, Uchanski RM, et al. Human discrimination of rotational velocities. *Exp Brain Res*. 2010;204:11–20.
- [31] Naseri AR, Grant PR. Human discrimination of translational accelerations. *Exp Brain Res*. 2012;218:455–464.
- [32] Soyka F, Robuffo Giordano P, Beykirch KA, et al. Predicting direction detection thresholds for arbitrary translational acceleration profiles in the horizontal plane. *Exp Brain Res*. 2011;209:95–107.
- [33] Soyka F, Giordano PR, Barnett-Cowan M, et al. Modeling direction discrimination thresholds for yaw rotations around an earth-vertical axis for arbitrary motion profiles. *Exp Brain Res*. 2012;220:89–99.
- [34] Samji A-A, Reid LD. The detection of low-amplitude yawing motion transients in a flight simulator. *IEEE Trans Systems, Man, Cybern*. 1992;22:300–306.
- [35] Groen EL, Bles W. How to use body tilt for the simulation of linear self motion. *J Vestib Res : Equilib Orientat*. 2004;14:375–385.
- [36] Valente Pais AR, Pool DM, Vroomede AM, et al. Pitch motion perception thresholds during passive and active tasks. *J Guid, Control, Dyn*. 2012;35:904–918.
- [37] Pool DM, Mulder M, Paassenvan MM, et al. Effects of peripheral visual and physical motion cues in roll-axis tracking tasks. *J Guid, Control, Dyn*. 2008;31:1608–1622.
- [38] Zaal PMT, Pool DM, Mulder M, et al. Multimodal pilot control behavior in combined target-following disturbance-rejection tasks. *J Guid, Control, Dyn*. 2009;32:1418–1428.
- [39] Drop FM, Pool DM, Damveld HJ, et al. Identification of the feedforward component in manual control with predictable target signals. *IEEE Trans Cybern*. 2013;43:1936–1949.
- [40] Nash CJ, Cole DJ. Identification of a novel model of driver steering control incorporating human sensory dynamics. Manuscript submitted for publication (2015).
- [41] Nash CJ, Cole DJ. Development of a novel model of driver-vehicle steering control incorporating sensory dynamics. In: Rosenberger M, Plöchl M, Six K, Edelmann J, editors. *The dynamics of vehicles on roads and tracks*. Graz, Austria: CRC Press; 2016. p. 57–66.
- [42] Nash CJ, Cole DJ. Measurement and Modelling of the Effect of Sensory Conflicts on Driver Steering Control. Manuscript submitted for publication (2016).
- [43] Timings JP, Cole DJ. Vehicle trajectory linearisation to enable efficient optimisation of the constant speed racing line. *Veh Syst Dyn*. 2012;50:883–901.
- [44] Timings JP, Cole DJ. Minimum maneuver time calculation using convex optimization. *J Dyn Syst, Meas Control*. 2013;135:e031015.
- [45] Perantoni G, Limebeer DJN. Optimal control for a formula one car with variable parameters. *Veh Syst Dyn*. 2014;52:653–678.
- [46] Falcone P, Borrelli F, Tseng HE, et al. Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *Int J Robust Nonlinear Control*. 2008;18:862–875.
- [47] Gerdtts M, Karrenberg S, Müller-Beler B, et al. Generating locally optimal trajectories for an automatically driven car. *Optim Eng*. 2009;10:439–463.
- [48] Erlien SM, Fujita S, Gerdes JC. Shared vehicle control using safe driving envelopes for obstacle avoidance and stability. *IEEE Trans Intell Transp Syst*. 2015;17:1–11.
- [49] Ungoren A, Peng H. An adaptive lateral preview driver model. *Veh Syst Dyn*. 2005;43:245–259.

- [50] Thommypillai M, Evangelou S, Sharp RS. Car driving at the limit by adaptive linear optimal preview control. *Veh Syst Dyn.* **2009**;47:1535–1550.
- [51] Keen SD, Cole DJ. Application of time-variant predictive control to modelling driver steering skill. *Veh Syst Dyn.* **2011**;49:527–559.
- [52] Hartikainen J, Solin A, Särkkä S. EKF/UKF Matlab toolbox.
- [53] Nash CJ, Cole DJ. Modelling the effect of sensory dynamics on a driver's control of a nonlinear vehicle. In: *Proceedings of the 13th international symposium on advanced vehicle control (AVEC 2016)*. Munich, Germany, 2016.
- [54] Donges E. A two-level model of driver steering behavior. *Human Factors: J Human Factors Ergon Soc.* **1978**;20:691–707.
- [55] Wills A, Ninness B. QPC - Quadratic Programming in C.
- [56] Hess RA. Analyzing manipulator and feel system effects in aircraft flight control. *IEEE Trans Syst, Man, Cybern.* **1990**;20:923–931.
- [57] Pick AJ, Cole DJ. Dynamic properties of a driver's arms holding a steering wheel. *Proc Instit Mech Eng, Part D: J Automob Eng.* **2007**;221:1475–1486.
- [58] Telban RJ, Cardullo F. Motion cueing algorithm development: human-centered linear and nonlinear approaches. NASA, 2005.
- [59] Pacejka HB, Bakker E. The magic formula tyre model. *Veh Syst Dyn.* **1992**;21:1–18.
- [60] Johns TA, Cole DJ. Measurement and mathematical model of a driver's intermittent compensatory steering control. *Veh Syst Dyn.* **2015**;53:1811–1829.