



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



**CC creative commons**  
COMMONS DEED

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

**BY:** **Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Noncommercial.** You may not use this work for commercial purposes.

**No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.


**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

# Evaluation of a new body-sideslip-based driving simulator motion cueing algorithm

Proc IMechE Part D:  
*J Automobile Engineering*  
226(11) 1433–1444  
© IMechE 2012  
Reprints and permissions:  
sagepub.co.uk/journalsPermissions.nav  
DOI: 10.1177/0954407012445247  
pid.sagepub.com  


**Nikhil JI Garrett and Matthew C Best**

## Abstract

This paper describes a new motion cueing algorithm for motion-based driving simulators. The algorithm uses the simulated vehicle's body sideslip angle as the demand for the motion platform's yaw degree of freedom. The current state of the art for motion cueing algorithms involves some form of filter or controller that limits the bandwidth of the vehicle motion before using this as the motion platform demand; the algorithm is tuned such that the platform does not exceed its limits. However, this means that information about the vehicle state that is contained within the motion is removed indiscriminately. Since the body sideslip angle will fit within the platform yaw limit under normal conditions, it does not need to be filtered beforehand, and thus no information must be removed. The implementation of the body-sideslip-based algorithm is described, as is a set of tests using human participants wherein the body sideslip algorithm was compared against the three most popular existing algorithms (namely the classical, adaptive and linear quadratic regulator algorithms) for normal road driving. The results of these tests indicate that the body sideslip algorithm performs as well as, or marginally better than, the other algorithms; future work will test the algorithm under limit handling conditions, to see whether the approach of preserving vehicle state information improves the simulator driver's perception.

## Keywords

Driving simulator, motion cueing, driver-in-the-loop, human factors

Date received: 18 August 2011; accepted: 16 March 2012

## Introduction

Driving simulators (in particular, those with motion) are increasingly popular in the fields of vehicle development and driver training. However, making use of motion to improve simulator realism is not a straightforward exercise. The problem arises from the characteristics of the motion platforms used, and specifically the limited motion workspace. Most simulators, including that at Loughborough University, use a Stewart-type platform with six degrees of freedom. The Loughborough simulator is a typical example of the type most popular in research and industry; the motion limits are of the order of  $\pm 0.5$  m in translation and  $\pm 20^\circ$  in rotation. Note that these limits are for motion on a single axis; motion on more than one axis will reduce the possible excursion in each. The available motion workspace is far smaller than would be required to reproduce fully the motion of a road vehicle in most normal manoeuvres, and this is even more true for the case of high-speed or racing-type driving. The problem therefore is how to transform the simulated

vehicle motion into a set of motion commands (or 'cues') that will fit within the workspace while by some measure provide the best possible feedback to the driver.

Previous work on the Loughborough simulator indicates that motion is an important factor in the perceived quality of the simulator;<sup>1</sup> in fact, motion is more important than steering feedback, peripheral screens and audio quality. The work of Siegler et al.<sup>2</sup> has shown that simulator motion prompts drivers to brake and corner as they would in real life; without motion, the subjects braked unusually hard and took unconventional lines around corners.

---

Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, UK

### Corresponding author:

Nikhil JI Garrett, Department of Aeronautical and Automotive Engineering, Stewart Miller Building, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK.  
Email: N.J.I.Garrett@lboro.ac.uk

Traditionally, the so-called classical algorithm uses linear high-pass filters to remove the unachievable steady state motion, reproducing the transient accelerations only.<sup>3</sup> Low-frequency lateral and longitudinal motion is often replayed on the roll and pitch degrees of freedom via low-pass filters, a process known as tilt coordination. The filters are tuned for a worst-case motion scenario; therefore, there is no possibility that the simulator reaches its motion limits. The advantage of this approach is that it is simple to implement, but the need to tune for a worst case means that the motion amplitude for most manoeuvres will be quite small.

A development of the classical algorithm uses adaptive filters.<sup>4</sup> The low-pass filters are left as linear filters with adaptive gains applied to the high-pass filters only. The adaptive gains, which are varied at each time step to minimise a cost function including motion error, platform excursion and departure from nominal gains, were found to reduce so-called false cues. These false cues are defined as platform motion with the opposite sign to the current vehicle motion. However, the tuning problem is even worse with this algorithm, with both filter parameters and cost function weightings to be chosen.

Subsequent cueing developments move away from simple filtering and towards optimal control-based approaches. The first and most widely quoted of these is based on linear quadratic regulator (LQR) design as proposed by Sivan et al.<sup>5</sup> The LQR approach designs a tracking controller whose aim is to track the perceived motion in the vehicle with the perceived motion in the simulator. In order to evaluate the perceived motion, models of the driver's vestibular system are included here; the dynamics of the motion platform are also modelled. As with previous algorithms the lateral-roll and longitudinal-pitch pairs are treated together. The controller is designed to minimise a cost function of perception error, platform positions and velocities, and acceleration demand. As it turns out, the resulting controller has very similar characteristics to the filter-based algorithms, in that the transfer functions again have a high-pass response. The effect of the platform roll on lateral motion perception is included in the vestibular model; therefore, a transfer function from the translational input to the rotational output also results and has a low-pass response. Tuning of this algorithm is perhaps slightly easier for a non-expert to carry out, as the cost function weightings have at least some obvious link to the motion itself.

A recent development uses model-based predictive control (MPC). The greatest advantage claimed for this approach is that constraints (in this case the motion platform limits) are included explicitly in the controller design and thus workspace usage is improved. This algorithm is unusual in that much of the development work was done on driving simulators instead of flight simulators. The algorithm designed by Dagdelen et al.<sup>6</sup> is such that the platform follows the vehicle motion for as long as possible and then switches to a washout

mode when it recognises that a limit will be reached. It would appear from the response of this algorithm that the sudden switching between modes would feel odd, and indeed this seems to be the case in the quoted results. A more conventional approach has been implemented on the Chalmers University of Technology driving simulator;<sup>7</sup> as yet, no results with a range of human subjects appear to have been published.

Results of comparisons between algorithms are rare in the literature; Telban et al.<sup>8</sup> presented a fairly thorough comparison for flight simulation. For driving simulation, results tend to be limited to objective analysis of time histories (see, for example, the papers by Nehaoua et al.,<sup>9</sup> Elloumi et al.<sup>10</sup> and Han et al.<sup>11</sup>). A more thorough investigation using subjective results has been presented by Grant et al.;<sup>12</sup> here, paired comparison techniques are used to evaluate which cueing algorithm is best for a double-lane-change manoeuvre, albeit for only two algorithms (classical and a 'lane-change' cue).

This paper proposes a new approach to the motion cueing problem. In general, the focus of previous algorithm development has been on matching (perceived) global vehicle motion as closely as possible within the platform workspace. Little attention has been paid to the information in the vehicle state relative to its own inertial frame. This is odd because vehicle models are always described in terms of these states, and they provide a direct measure of lateral stability in the form of the body sideslip angle, i.e. the yaw attitude of the body relative to the total velocity vector. For all but the most extreme cases of loss of control, the sideslip angle is bounded within the platform's yaw limits, and so there is no need to apply any arbitrary filter. Given that it is not possible to reproduce absolute yaw accelerations of magnitude and duration anywhere near those in the real vehicle, it seems almost futile to try; by applying body sideslip cues instead, the focus is placed on providing important information about the vehicle's state of stability to the driver. In existing cueing approaches, the heavy filtering of the vehicle yaw motion means that much of the yaw cueing comes from the visual system; in the new algorithm, the large-scale yaw rate is communicated through the visual system, while the lateral stability information is provided through the platform yaw motion.

It is anticipated that the approach described above will be most beneficial for driving in the non-linear region of vehicle behaviour, where information on the vehicle stability is critical. However, in order to establish the suitability of the body sideslip algorithm, it is also necessary to evaluate it for driving in the linear region of vehicle behaviour. Some tests were carried out to this end; the results are presented here.

The Loughborough simulator, including the vehicle dynamics model, is described in the second section. The third section presents the body sideslip cueing algorithm including the method by which the derivatives of the sideslip angle are obtained from the vehicle dynamics;

the fourth section details the comparator cueing algorithms. The fifth section describes the test procedure for testing the body sideslip algorithm against three other algorithms with human drivers; the results of the tests are presented and discussed in the sixth section.

**Simulator details**

The Loughborough driving simulator was supplied by Cruden<sup>13</sup> and is based around a Moog ECUE624-1800 electric motion platform. The platform is a Stewart–Gough-type platform, the six actuators each having a stroke of 24 in (610 mm). The graphics are displayed on three PC monitors in front of the driver. A d.c. motor provides torque feedback through the steering, the accelerator pedal has a simple spring loading, and the brakes use a hydraulic system to provide realistic feel.

An additional feature worth noting is the ability, via an application running on an extra PC connected to the simulator local network, to send signals into the vehicle model during the run time. This can be used to switch between different vehicle parameters in real time or, as used here, to switch between motion cueing algorithms. This is of great use during paired comparison tests, as two cueing algorithms can be tested back to back without having to stop and start the simulation.

The vehicle dynamics model is implemented in MATLAB/Simulink and runs with a fixed integrator step size of 1 ms. The model is compiled using MATLAB Real-Time Workshop. One of the outputs of the compiled model is the motion platform command which consists of the position, velocity and acceleration signals for each of the six degrees of freedom; a low-level controller then calculates a set of actuator currents.

The tyre model is a combined-slip Pacejka ‘magic formula’ model (see the papers by Bakker et al.<sup>14,15</sup> for the original model proposal and the book by Pacejka<sup>16</sup> for a detailed description) and is largely based on the implementation by Milliken and Milliken.<sup>17</sup> Additionally, elements of the tyre model given by Dixon<sup>18</sup> are included, specifically the variations in the cornering stiffness and the friction coefficient with the load.

The slip ratio  $s_x$  and slip angle  $\alpha$  are normalised, and a combined slip quantity  $s_r$  calculated from

$$\bar{s}_x = \frac{C_x s_x}{F_{x,peak}} \tag{1}$$

$$\bar{\alpha} = \frac{C_a \alpha}{F_{y,peak}} \tag{2}$$

$$s_r = \sqrt{\bar{s}_x^2 + \bar{\alpha}^2} \tag{3}$$

The magic formula then calculates the tyre force as

$$F_r = \sin\left(C \arctan\left\{\frac{s_r}{C} - E\left[\frac{s_r}{C} - \arctan\left(\frac{s_r}{C}\right)\right]\right\}\right) \tag{4}$$

The tyre stiffnesses and friction coefficient are all dependent on the vertical load according to

$$C_a = c_1 z_{rd} \left(1 - e^{-c_2(F_z/z_{rd})}\right) \tag{5}$$

$$C_x = C_a c_{fac} \tag{6}$$

$$\mu' = \mu \left(4 \frac{F_z}{z_{rd}}\right)^{-0.15} \tag{7}$$

Tyre relaxation is modelled by a simple first-order lag function

$$\hat{F}_{xi,yi} = \tau^{-1}(F_{xi,yi} - \hat{F}_{xi,yi}) \tag{8}$$

The tyre model also calculates the pneumatic trail for use in the calculation of steering feedback torque in the vehicle model as

$$t_{pneu} = t_{max} \cos\left(C \arctan\left\{\frac{s_r}{C} - E\left[\frac{s_r}{C} - \arctan\left(\frac{s_r}{C}\right)\right]\right\}\right) \tag{9}$$

The vehicle model topology is illustrated in Figure 1. The rotational dynamics of the wheel are governed by a simple drivetrain model. The steering torque is calculated as the sum of trail, geometric and lateral contact patch offset terms according to

$$M = F_{y,front}(t_{pneu} + t_{caster}) + M_{geo} + M_{latoff} \tag{10}$$

The numerical values for the quantities mentioned above and in subsequent sections are included in Appendix 2.

**Body sideslip cueing algorithm**

The idea behind the sideslip cueing algorithm is that the body sideslip angle is used as the motion platform’s yaw demand. In order to provide full position, velocity and acceleration information to the platform controller, the

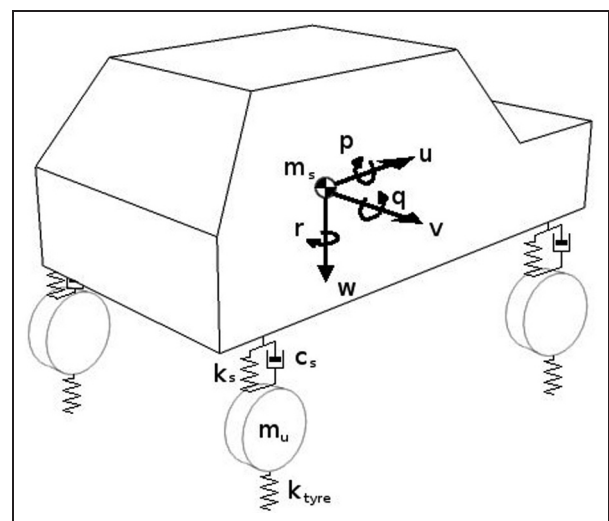


Figure 1. Illustration of the vehicle dynamics model.

first and second derivatives of the body sideslip angle are required.

The sideslip angle itself is readily available from the vehicle model as  $\beta = \arctan(v/u)$ ; since the sideslip angle is generally small (less than  $10^\circ$ ), a small-angle approximation

$$\beta = \frac{v}{u} \quad (11)$$

can be used. At larger angles, this assumption introduces some error, but the platform can only provide a maximum  $20^\circ$  yaw angle, at which the error is only 4%. Since a sideslip angle above  $20^\circ$  is very rare and is also associated with loss of control of the vehicle, the small-angle approximation is appropriate.

The first derivative given by

$$\dot{\beta} = \frac{\dot{v}}{u} - \frac{\dot{u}v}{u^2} \quad (12)$$

can also be directly calculated from the model states; however, initial examination of the second derivative

$$\ddot{\beta} = \frac{\ddot{v}}{u} - \frac{\ddot{u}v}{u^2} - 2\frac{\dot{u}\dot{v}}{u^2} + 2\frac{\dot{u}^2v}{u^3} \quad (13)$$

suggests a problem, i.e. that there are jerk terms  $\ddot{u}$  and  $\ddot{v}$ ; these terms are not directly available from the vehicle dynamics.

However, examining the expressions for  $a_x$  and  $a_y$ , which are

$$a_x = \frac{\sum F_x}{m} = \dot{u} - vr - wq \quad (14)$$

$$a_y = \frac{\sum F_y}{m} = \dot{v} + ur + wp \quad (15)$$

and differentiating with respect to time to give

$$\frac{\sum \dot{F}_x}{m} = \ddot{u} - \dot{v}r - v\dot{r} - \dot{w}q - w\dot{q} \quad (16)$$

$$\frac{\sum \dot{F}_y}{m} = \ddot{v} + \dot{u}r + u\dot{r} + \dot{w}p + w\dot{p} \quad (17)$$

we now have expressions for  $\ddot{u}$  and  $\ddot{v}$  according to

$$\ddot{u} = \frac{\sum \dot{F}_x}{m} + \dot{v}r + v\dot{r} + \dot{w}q + w\dot{q} \quad (18)$$

$$\ddot{v} = \frac{\sum \dot{F}_y}{m} - \dot{u}r - u\dot{r} - \dot{w}p - w\dot{p} \quad (19)$$

The velocities and their derivatives are all available as vehicle states; the only unknowns here are the tyre force derivatives. However, examination of the expression for relaxation lag (equation (8)) reveals that the derivatives are in fact available from the tyre model. Thus the jerk terms required for the body sideslip acceleration can be calculated.

For the body sideslip cueing, the body sideslip angle provides the platform's yaw demand. Some lateral and roll motion are also added; the lateral displacement was simply the body sideslip angle multiplied by a scale

factor, and the simulated vehicle roll was used as the platform's roll demand. The lateral motion is intended to move the centre of rotation away from the platform's centroid (which is directly below the driver's head); a future algorithm development will have a moving rotation centre dependent on the vehicle state. Initial testing with unity gain applied to the yaw and roll angles indicated that drivers felt that there was far too much yaw and roll motion. This is perhaps explained by the resulting mismatch in magnitude between the unfiltered yaw and roll and the filtered (and attenuated) motion in other degrees of freedom. It was thus decided to apply a scale factor to the yaw and roll demand to bring them down to levels acceptable to drivers.

## Comparator cueing algorithms

The body sideslip cueing is tested against the classical, adaptive and LQR algorithms; the motion in the lateral, roll and yaw degrees of freedom is applied using the algorithms under test. In order to achieve a fair comparison the longitudinal, pitch and vertical motions are applied using Cruden's default classical cueing algorithm. The tilt coordination gain for the pitch-longitudinal pair was set to zero to prevent large pitch motion. This set-up then provides motion in all six axes but the test cases are focused on the motion associated with cornering.

### Classical and adaptive algorithms

The classical and adaptive implementation is based on that described by Nahon et al.<sup>19</sup> The filter topology for a tilt-coordinated pair (e.g. lateral and roll) is shown in Figure 2.

The high-pass filter for lateral acceleration has the form

$$G_y(s) = \frac{Ps^3}{s^3 + (2\zeta\omega_1 + \omega_2)s^2 + (\omega_1^2 + 2\zeta\omega_1\omega_2)s + \omega_1^2\omega_2} \quad (20)$$

The high-pass filters for roll and yaw are of the form

$$G_{r,p}(s) = \frac{Ps^2}{s^2 + 2\zeta\omega s + \omega^2} \quad (21)$$

In equations (20) and (21), the filter gain  $P$  is set to 1 for the classical algorithm. For the adaptive algorithm

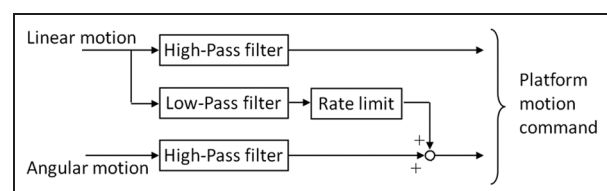


Figure 2. Filter topology for the classical algorithm.



the gains are varied during simulation to minimise the cost function; the initial value of this gain is 1 in all cases.

The cost function for the lateral–roll pair is

$$J_y = \frac{1}{2} [W_1(a_y - \ddot{y}_s)^2 + W_2(\dot{\phi} - \dot{\phi}_s)^2 + W_3\dot{x}_s^2 + W_4x_s^2 + W_5\dot{\phi}_s^2 + W_6\phi_s^2 + W_7(P_y - P_{y0})^2 + W_8(P_\phi - P_{\phi0})^2] \quad (22)$$

where  $P_{i0}$  are the initial values for the adaptive gains. The yaw cost function is similar but with the rotational terms only. The adaptive gains are varied at each time step by the steepest-descent method according to

$$\dot{P} = -K \frac{\partial J_y}{\partial P} \quad (23)$$

The tilt coordination low-pass filter has the form

$$G_{lp}(s) = \frac{\omega_1^2 \omega_2}{s^3 + (2\zeta\omega_1 + \omega_2)s^2 + (\omega_1^2 + 2\zeta\omega_1\omega_2)s + \omega_1^2\omega_2} \quad (24)$$

Note that this filter does not have an adaptive gain; the filter remains linear throughout.

## LQR

The LQR cueing is based on the implementation by Sivan et al.<sup>5</sup> For the lateral–roll pair, a single transfer function is produced that has two inputs and two outputs; a single-input–single-output transfer function is produced for the yaw degree of freedom. The formulation is a tracking LQR problem, where the perceived motion in the simulator should track the perceived motion in the real vehicle.

The vestibular system is modelled using second-order transfer functions for the body's translational and rotational motion transducers, the otoliths and the semicircular canals. The otolith transfer function from the head's acceleration  $a$  to the perceived acceleration  $\hat{a}$  is

$$\frac{\hat{a}}{a} = \frac{K_{oto}(\tau_n s + 1)}{(1 + \tau_{SS})(1 + \tau_L s)} \quad (25)$$

The semicircular canal transfer function from the head's angular velocity  $\omega$  to the perceived angular velocity  $\hat{\omega}$  is

$$\frac{\hat{\omega}}{\omega} = \frac{K_{scc}s}{(1 + \tau_a s)(1 + \tau_2 s)} \quad (26)$$

Note that the semicircular canals function as angular velocity transducers, and the otoliths as linear acceleration transducers. The motion platform dynamics are modelled as a first-order low-pass filter with a cut-off frequency of 20 Hz.

The vestibular and platform dynamics are combined into a single state-space model, such that a controller can be found that minimises the cost function  $J$  given by

$$J = E\{e^T Q e + x_d^T R_d x_d + u_s^T R u_s\} \quad (27)$$

where  $e$  is the perception error,  $x_d$  is the vector of platform positions and velocities,  $u_s$  is the control input to the platform, and  $Q$  and  $R$  are the matrices of cost weightings. With some rearrangement, a washout filter  $W(s)$  is found that transforms the simulated vehicle states into a set of platform demands,  $u_s = W(s)u_{vehicle}$ . For the lateral–roll pair, the washout filter can be separated into its four component transfer functions according to

$$W(s) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \quad (28)$$

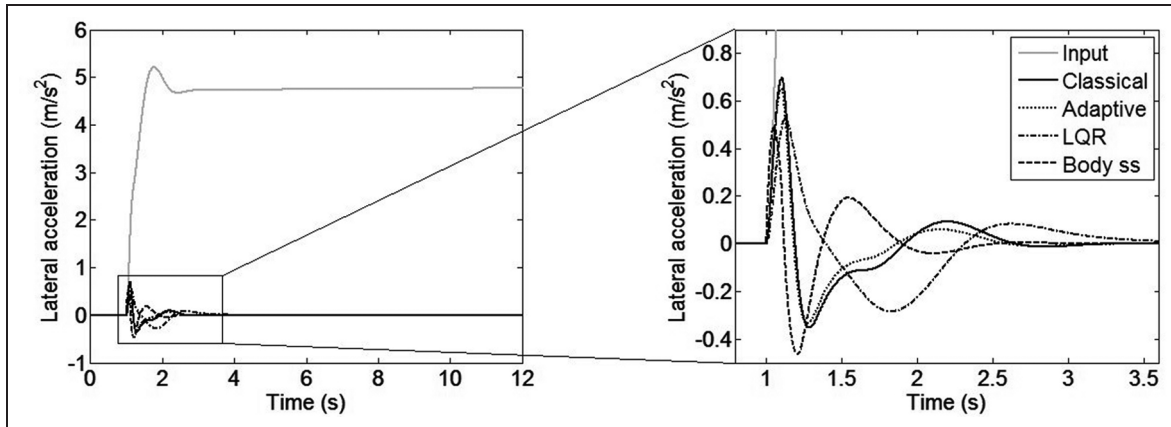
As mentioned in the Introduction, examination of  $W_{11}$  and  $W_{22}$  shows that these end up being high-pass filters as in the classical and adaptive algorithms, the difference being that these are of higher order (fifth order instead of third order). The transfer function  $W_{21}$  from the lateral input to the roll output turns out to have a low-pass characteristic; therefore, as in the classical and adaptive algorithms, the low-frequency lateral motion is reproduced using roll motion.

## Tuning

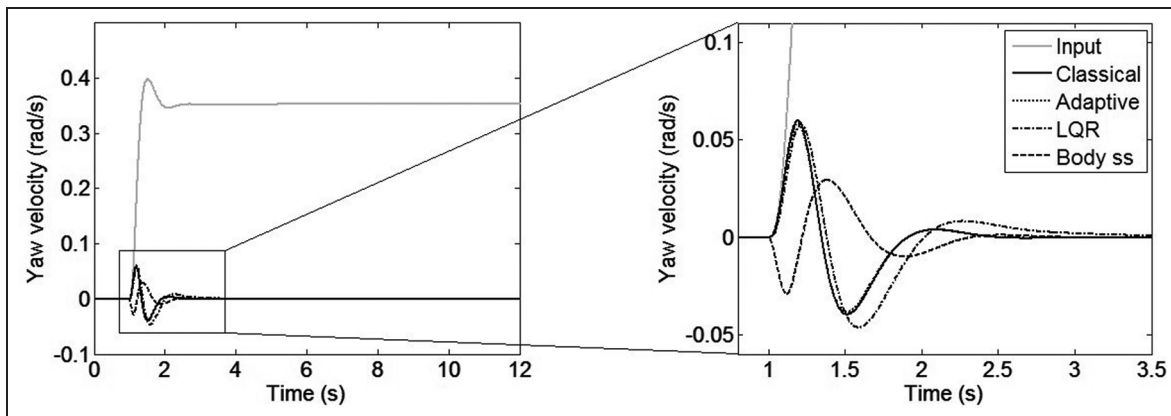
As mentioned in the Introduction, tuning of the motion cueing algorithms can prove time consuming and tends to require both an experienced driver and an experienced operator. An additional concern is that the tuning will significantly affect the drivers' perception. A method is sought to tune the algorithms using an objective performance metric. The standard tuning procedure starts by ensuring that the maximum platform displacement is within the platform limits for a worst-case manoeuvre. One possibility for tuning would be to ensure that this maximum displacement is the same for all algorithms; however, given the different behaviour of the algorithms, this is not sensible. Instead, the r.m.s. values are matched. Given that the body is sensitive to linear accelerations and angular velocities, the r.m.s. values for these quantities were matched across all algorithms.

The r.m.s. values for the classical algorithm were chosen as the reference parameters, and the other three algorithms were tuned to match them. The filter cut-off frequencies and damping for the classical and adaptive filters were chosen on the basis of those used in Cruden's standard cueing algorithm. The initial adaptive filter cost function weightings were taken from the work of Nahon et al.<sup>19</sup> The initial LQR weightings were taken from the work of Sivan et al.<sup>5</sup> The tuning was based on a recorded lap of the test course. Figure 3 and Figure 4 are the lateral acceleration and yaw velocity responses respectively of the simulated vehicle (labelled Input) and the tuned algorithms for a 5° step-steer manoeuvre at 30 mile/h.

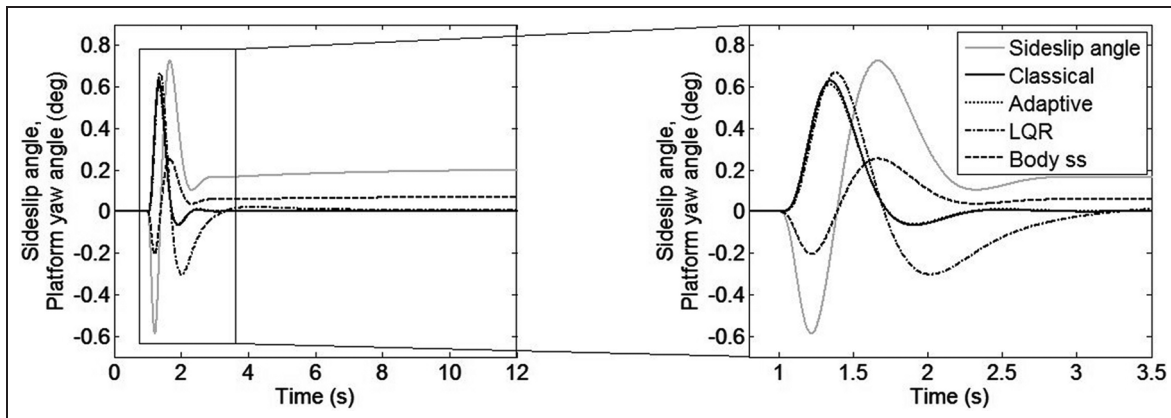
As mentioned earlier, the classical, adaptive and LQR algorithms all have a high-pass response. The



**Figure 3.** Lateral acceleration response to a  $5^\circ$  step steer.  
LQR: linear quadratic regulator; ss: sideslip.



**Figure 4.** Yaw velocity response to a  $5^\circ$  step steer.  
LQR: linear quadratic regulator; ss: sideslip.



**Figure 5.** Body sideslip angle and platform's yaw response for a  $5^\circ$  step steer.  
LQR: linear quadratic regulator; ss: sideslip.

body sideslip algorithm also has a high-pass-type response in the lateral degree of freedom. In yaw, the initial velocity of the sideslip algorithm has the opposite sign to the other algorithms (and to the vehicle's yaw velocity). Examination of the platform yaw angle

demand, plotted together with the vehicle's sideslip angle in Figure 5, reveals that for this particular manoeuvre the body sideslip angle is initially negative.

Figure 5 illustrates that, unlike the other three algorithms, the body sideslip algorithm response is in phase



**Figure 6.** View from above the simulator driver's seat.

with the vehicle's sideslip angle and has the same steady state characteristic for this manoeuvre.

**Trial conditions**

The tests were based on bidirectional paired comparisons between the four cueing algorithms, i.e. each possible pair was tested in both directions. The order of the pairs was also randomised for each test subject. Thus, 13 laps are necessary to carry out all possible permutations of the four algorithms; it was judged that a continuous block of 13 laps would be too long to expect the participants to maintain concentration, and so a mid-session break was included. To avoid a situation in which participants have to remember an algorithm across the break, an extra lap was added after the break using the same algorithm as the last lap before the break. This took the total number of laps to 14, i.e. two blocks of seven.

22 participants took part in the tests, with eight from the 18–30 age group, seven from the 31–60 age group, and seven from the 60 and over age group. The male–female split was approximately 50:50, and as far as possible there was a range of experience levels, both of real-world and simulator driving. For five of the participants the test had to be stopped owing to simulator sickness; also one of the participants did not understand the test properly and so the results for this participant were discarded.

A mountain road course was selected for its high number of corners and range of low- to medium-speed corners, uphill and downhill sections, and crossfall and

adverse camber; Figure 6 shows the view of the road from just above the driver's head. The nature of the course made most participants drive in the linear region without prompting; it is also a good choice for detecting motion sickness, as those participants who did suffer from motion sickness noticed within the first lap and the remainder did not experience any sickness or stress for the length of the trial. The length of the course (1 mile; average lap time, around 2:30–3:00 min) provided drivers with sufficient time to judge the motion cueing without making the test period too long.

The vehicle parameters are those of a large saloon car, namely the Jaguar XJ. These parameters were taken where possible from the manufacturer's data; the tyre parameters were tuned to match the understeer characteristics of a test vehicle. The car was simulated with an automatic transmission to reduce the cognitive load on the drivers and to allow them to concentrate on vehicle handling and motion cueing.

All participants drove five laps to familiarise themselves with the vehicle and the test route and to find a comfortable pace (they were asked to drive as they would on a public road); the classical cueing algorithm was selected on the test laps. In the cases where drivers experienced motion sickness, the tests were aborted and the drivers excluded from the study.

On each lap the participants were asked to pull away, to complete the lap at the pace that they found comfortable and then to bring the car to a stop. After each lap the cueing algorithm was changed by the operator. From the second lap onwards in each set of seven, the participants were asked how realistic the simulator felt on the lap that they had just driven compared with the previous lap, using a Likert scale (Figure 7).

The participants were asked to concentrate on only the two most recent laps, and not to worry about how they compared with any earlier laps. They were told that a change was being made to the way that the platform moved, but not which degrees of freedom would be changed or the type of change.

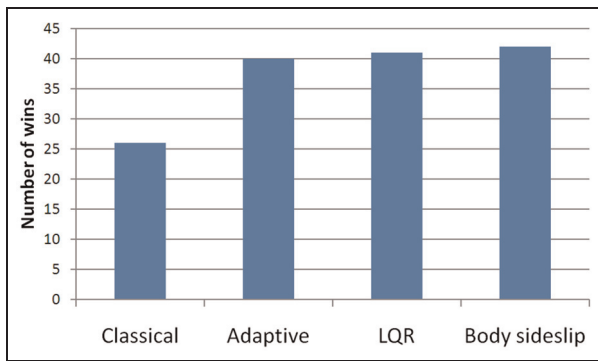
**Results and discussion**

The tests described above generate a series of preferences between cueing algorithms for each participant. Perhaps the simplest and most intuitive way to present these results is simply to count the number of 'wins' for each algorithm, i.e. the number of times that each algorithm was preferred to another (for those tests where the response was 'about the same', neither algorithm

How realistic was the simulator compared with the previous lap?				
Much worse	Worse	About the same	Better	Much better
-2	-1	0	+1	+2

**Figure 7.** Likert scale for simulator tests.





**Figure 8.** Number of wins for each algorithm.  
LQR: linear quadratic regulator.

records a ‘win’). The result of this analysis is presented in Figure 8.

These results suggest that the adaptive, LQR and body sideslip algorithms are comparable in terms of how realistic they are, with the classical algorithm coming out slightly worse.

A more formal statistical evaluation of the results is carried out using the Friedman test;<sup>20</sup> this is a non-parametric form of a two-way analysis-of-variance (ANOVA) test. The choice of the Friedman test over ANOVA was made because ANOVA requires that the data are drawn from a normal distribution; this is not necessarily the case here. The Friedman test is based on ranks; for each block the items under test are assigned a value from 1 to  $n$  (where  $n$  is the number of test items) according to the order of preference; tied rankings are assigned mean values. The Friedman statistic  $Q$  is then calculated using

$$Q = \frac{12N}{s(s+1)} \sum_{i=1}^s \left[ R_i - \frac{1}{2}(s+1) \right]^2 \quad (29)$$

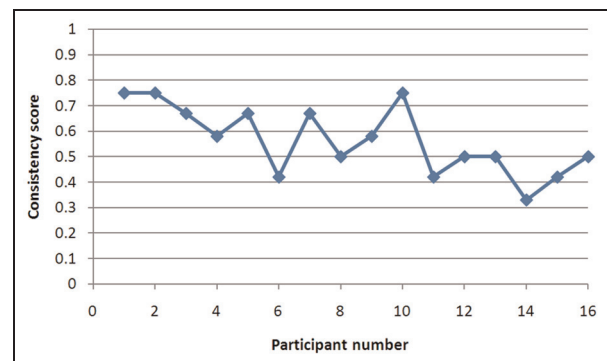
where  $N$  is the number of blocks,  $s$  is the number of items under test and  $R_i$  are the rankings for each block. The probability that the null hypothesis (i.e. that the  $R_i$  are equal) can be rejected is found from a table of  $Q$  values.

For each participant the six possible pairings of the four algorithms were evaluated in the forward direction and the reverse direction. The forward results and the reverse results for each participant are ranked separately and form a block in the Friedman test; when evaluating the algorithms against each other, we wish to ignore any effects from differences between the blocks (participants). The consistency of the participants is discussed later. The Friedman test results indicate that the probability that the null hypothesis (all algorithms are equally good) is true is 0.09. Although a probability below 0.05 is often quoted (see, for example, the book by Noether<sup>21</sup>) as the threshold for rejecting the null hypothesis, this still indicates to a reasonable degree that the results are indeed due to differences between the algorithms rather than to random effects.

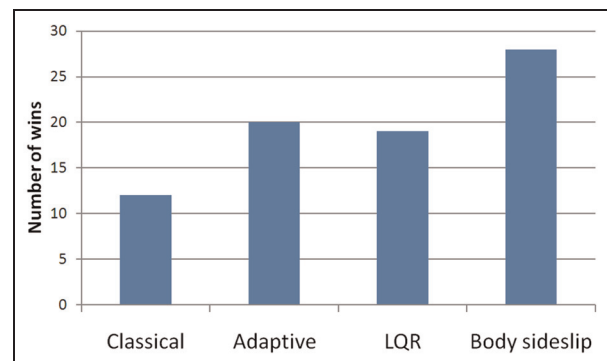
Since these paired comparisons were made in both directions, it is also possible to check whether participants were consistent in their two results for each pair. A simple measure of consistency was calculated as follows; for each pair of algorithms, a participant was scored a 1 if their preference was the same in both directions, 0.5 if they had a preference in one direction but said the algorithms were the same in the other, and 0 if they gave preference to a different algorithm in each direction. This score was then averaged over the six pairs. The results are shown in Figure 9.

If we choose the eight participants with an average score greater than 0.5 and recalculate the earlier results, we see that the probability from the Friedman test is reduced to 0.06. The ranking formed by counting the number of ‘wins’ is shown in Figure 10; note that the general trend is similar but the body sideslip algorithm now seems to be slightly preferred to the adaptive and LQR algorithms. We should be cautious about the results here, however; the original sample size of 16 is small and, after narrowing this to eight, we should regard conclusions as tentative rather than firm.

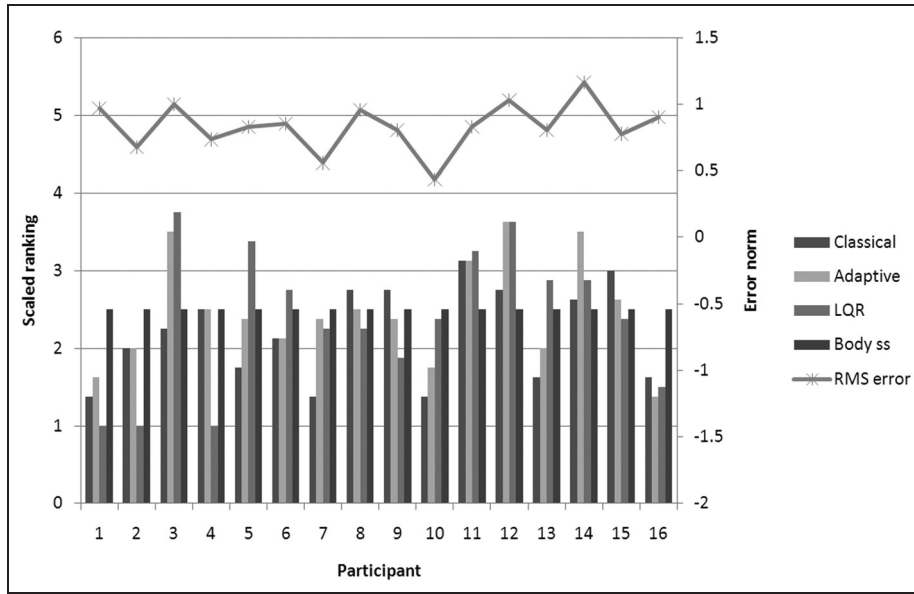
An alternative analysis uses a model-fitting approach; a least-squares regression is used to find the best-fit ranking for each participant. The results of each comparison can be recorded in matrix form



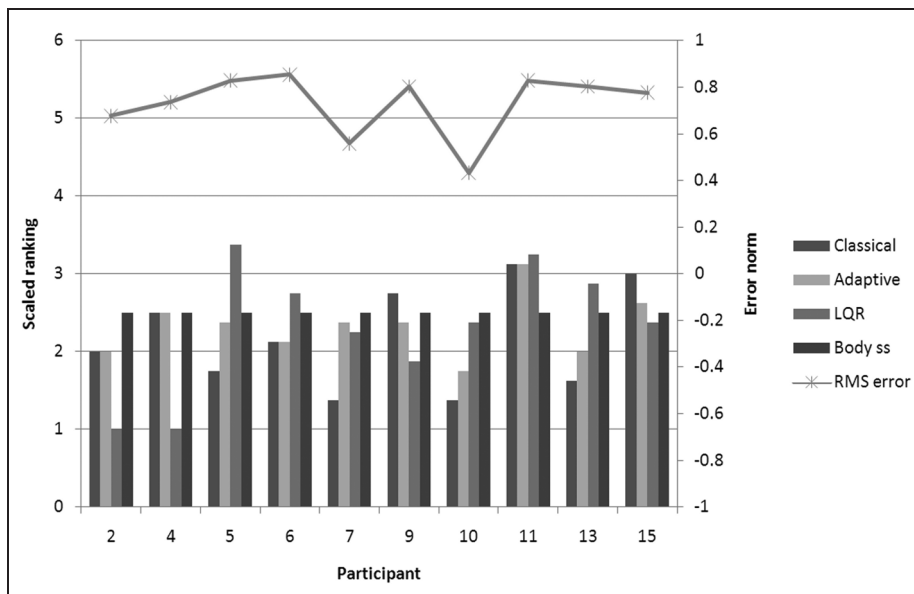
**Figure 9.** Consistency scores for 16 participants.



**Figure 10.** Number of wins for the eight most consistent participants.  
LQR: linear quadratic regulator.



**Figure 11.** Regression results and error for each participant. LQR: linear quadratic regulator; ss: sideslip.



**Figure 12.** Results for the 10 most consistent participants. LQR: linear quadratic regulator; ss: sideslip.

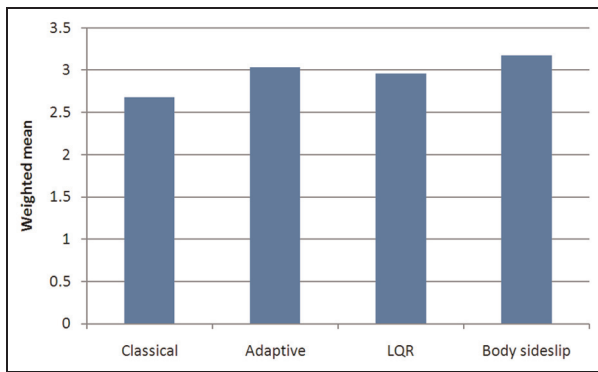
$$U\theta = y$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_{cl} \\ \theta_{ad} \\ \theta_{lqr} \\ \theta_{ss} \end{bmatrix} = \begin{bmatrix} +1 \\ -2 \\ \vdots \\ -1 \\ 0 \end{bmatrix} \quad (30)$$

where the matrix  $U$  represents the different comparisons (the first two rows shown above are the comparison between the classical and adaptive algorithms in both directions), and the vector  $y$  contains the results

of the comparisons. A least-squares regression finds an estimate  $\hat{\theta}$  of the vector  $\theta$  that best fits the results.

The error vector  $e = U\hat{\theta} - y$  is an alternative indicator of consistency applicable to the results. Figure 11 shows  $\hat{\theta}$  for each participant, together with the r.m.s. value for the error vector. Note that the calculated  $\hat{\theta}$  values have all had the same scalar value added to them to make the plot clearer; this does not affect the ranking of the algorithms. Also note that the outcome of the regression has the body sideslip as the reference; hence this is why the ‘score’ is the same for the body sideslip across all participants.



**Figure 13.** Weighted mean of the least-squares rankings. LQR: linear quadratic regulator.

These results give slightly more insight into the variability between subjects; it is clear from Figure 11 that there is significant noise in the results. Looking at those participants with an r.m.s. error norm less than 0.8 (i.e. excluding the six least consistent participants, with the results repeated in Figure 12 for clarity), it could again be concluded that the classical algorithm is generally rated worse than, or as good as, the other algorithms.

An overall result from this analysis is shown in Figure 13. This is the mean  $\hat{\theta}$  value for each algorithm, weighted by the inverse of the r.m.s. error. As with the results in Figure 8, the adaptive, LQR and body sideslip algorithms all have similar rankings; however, the classical algorithm is closer to the performance of the other three here.

A pessimistic interpretation of these results suggests that there is no clear preference for any of the four algorithms under test; there is some inconsistency in all the assessors' responses and the sample size is quite small. More positively, all the average measures show that the body sideslip method is the best algorithm, and the results in Figure 10 demonstrate that, when concentrating on the eight most consistent subjects, the margin of preference is significant. In the context of the research overall, this is a promising result; we should bear in mind that, since sideslip is principally associated with vehicle stability, the expectation is that the most significant benefit of the body sideslip algorithm will be seen in limit handling. If this prediction is correct, the fact that the algorithm is at least on a par with the three most commonly used algorithms for driving in the linear region means that this cueing should improve overall simulator fidelity over a range of conditions.

As an additional note, some test subjects struggled to maintain concentration for the duration of the tests; future testing will therefore treat this test length as a maximum and shorten it slightly if possible. It is also interesting to note that, after the tests were completed, some participants made some comments about what they thought was being changed each time; some were surprised to find that there were only four different setups, and a surprising number were convinced that the longitudinal behaviour was also changing.

## Concluding remarks

The body sideslip algorithm was found to perform well against the three most popular motion cueing algorithms. The results appear to show that the body sideslip algorithm is as good as the adaptive and the LQR algorithms, and that all three provide a slight improvement over the classical algorithm. However, none of the algorithms stands out as being a clear choice for testing in the linear region of vehicle behaviour. It seems reasonable to conclude therefore that the body sideslip algorithm is as good as the other algorithms for simulation of normal road driving. Since the real benefit of the approach is expected for simulation of driving near the vehicle limit, this is a promising result; it means that, if the sideslip algorithm does turn out to be an improvement in limit handling manoeuvres, then it can be employed on simulators without affecting the quality of motion for manoeuvres away from the vehicle limit.

It is intended that the body sideslip cueing algorithm be developed further, particularly the lateral component of the motion, and evaluated against at least two other algorithms; one based on a non-linear formulation of LQR, generalised optimal control, which allows the use of non-linear system models and non-quadratic cost functions, and the other being the MPC-based algorithm described by Augusto and Loureiro.<sup>7</sup> Some testing will also be performed with so-called expert drivers; by this, we mean drivers who have experience of driving at (and indeed beyond) the limit of a vehicle's cornering capabilities. The idea behind these tests is to evaluate how good the body sideslip algorithm is at providing feedback about the vehicle state near the vehicle limits, and correspondingly whether the drivers feel that they can control the simulated vehicle as they would a real vehicle.

## Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

## References

1. Newton AP and Best MC. The influence of motion on handling dynamics analysis in full vehicle simulators. In: *8th international symposium on advanced vehicle control*, Taipei, Republic of China, 20–24 August 2006, pp. 303–308. Tokyo: JSAE.
2. Siegler I, Reymond G, Kemeny A and Berthoz A. Sensorimotor integration in a driving simulator: contributions of motion cueing in elementary driving tasks. In: *Driving simulation conference*, Nice, France, 5–7 September 2001, pp.21–32. Paris: IFSTTAR.
3. Schmidt SF and Conrad B. Motion drive signals for piloted flight simulators. NASA Contractor Report CR-1601, National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California, USA, 1970.
4. Parrish RV, Dieudonne JE, Bowles RL and Martin DJ. Coordinated adaptive washout for motion simulators. *J Aircr* 1975; 12(1): 44–50.

5. Sivan R, Ish-Shalom J and Huang JK. An optimal control approach to the design of moving flight simulators. *IEEE Trans Man, Systems, Cybernetics* 1982; 12(6): 818–827.
6. Dagdelen M, Reymond G, Kemeny A et al. Model-based predictive motion cueing strategy for vehicle driving simulators. *Control Engng Practice* 2009; 17(9): 995–1003.
7. Augusto BDC and Loureiro RJL. *Motion cueing in the Chalmers driving simulator: a model predictive control approach*. MSc Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2009.
8. Telban RJ, Cardullo FM and Kelly LC. Motion cueing algorithm development: piloted performance testing of the cueing algorithms. NASA Contractor Report CR-2005-213748, National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia, USA, 2005.
9. Nehaoua L, Arioui H, Espie S and Mohellebi H. Motion cueing algorithms for small driving simulators. In: *IEEE international conference on robotics and automation*, Orlando, FL, USA, 15–19 May 2006, pp.3189–3194. New York: IEEE.
10. Elloumi H, Bordier M and Maizi N. An optimal control scheme for a driving simulator. In: *2nd international conference on informatics in control, automation and robotics*, Barcelona, Spain, 14–17 September 2005, pp.40–47. Setubal: INSTICC Press.
11. Han MC, Lee HS, Lee S and Lee MH. Optimal motion cueing algorithm using the human body model. *JSME Int J Ser C* 2002; 45(2): 487–491.
12. Grant PR, Blommer M, Artz B and Greenberg J. Analysing classes of motion drive algorithms based on paired comparison techniques. *Veh System Dynamics* 2009; 47(9): 1075–1093.
13. Cruden, <http://www.cruden.com> (2011).
14. Bakker E, Nyborg L and Pacejka HB. Tyre modelling for use in vehicle dynamics studies. SAE paper 870421, 1987.
15. Bakker E, Pacejka HB and Lidner L. A new tire model with an application in vehicle dynamics studies. SAE paper 890087, 1989.
16. Pacejka HB. *Tyre and vehicle dynamics*. Oxford: Butterworth–Heinemann, 2002.
17. Milliken WF and Milliken DL. *Race car vehicle dynamics*. Warrendale, PA: SAE International, 1995.
18. Dixon JC. *Tires, suspension and handling*, 2nd edition. Warrendale, PA: SAE International, 1996.
19. Nahon MA, Reid LD and Kirdeikis J. Adaptive simulator motion software with supervisory control. *J Guidance, Control Dynamics* 1992; 15(2): 376–383.
20. Lehmann EL. *Nonparametrics: statistical methods based on ranks*. New York: McGraw-Hill, 1975.
21. Noether GE. *Introduction to statistics – the nonparametric way*. Berlin: Springer, 1991.

**Appendix I**

**Notation**

$a$	linear acceleration
$C_x, C_a$	longitudinal stiffness and lateral stiffness respectively of the tyres

$C, E$	Pacejka magic formula coefficients
$F_x, F_y$	longitudinal force and lateral force respectively of the tyres
$F_z$	vertical load on the tyre
$G$	filter transfer function
$I$	moment of inertia
$J$	cost function
$K$	adaptive step size, scalar gain
$m$	mass of the vehicle
$M$	steering torque
$p$	roll velocity
$P$	adaptive gain
$q$	pitch velocity
$Q$	Friedman statistic
$Q$	linear quadratic regulator cost function
	weight matrix
$r$	rolling radius of the wheel, yaw velocity
$R$	Friedman test rankings
$R$	linear quadratic regulator cost function
	weight matrix
$s$	tyre slip, Laplace operator, Friedman test
	number of items
$t$	trail
$u$	longitudinal velocity
$v$	lateral velocity
$W$	adaptive filter cost function weight
$W$	washout filter
$z_{rd}$	reference vertical load of the tyre
$\alpha$	slip angle of the tyre
$\beta$	body sideslip angle
$\zeta$	filter damping ratio
$\theta$	regression variable
$\mu$	friction coefficient
$\tau$	time constant
$\omega$	angular velocity, filter cut-off frequency

**Subscripts**

$oto$	otolith
$scc$	semicircular canal

**Superscripts**

$\wedge$	perceived quantity
----------	--------------------

**Appendix 2**

**Numerical values**

The numerical values for the vehicle model and the cueing algorithm parameters are given in Table 1.



**Table 1.** The vehicle model parameters and the cueing algorithm parameters.

Symbol	Value	Description
<b>Vehicle model parameters</b>		
$m_s$	1678 kg	Sprung mass of the vehicle
$m_u$	40 kg	Unsprung mass (per wheel)
$I_{xx}$	735 kg m <sup>2</sup>	Roll moment of inertia of the vehicle about the centre of gravity
$I_{yy}$	4000 kg m <sup>2</sup>	Pitch moment of inertia of the vehicle about the centre of gravity
$I_{zz}$	4136 kg m <sup>2</sup>	Yaw moment of inertia of the vehicle about the centre of gravity
$a$	1.39 m	Longitudinal distance from the front axle to the centre of gravity
$b$	1.64 m	Longitudinal distance from the rear axle to the centre of gravity
$t_f$	1.56 m	Front track
$t_r$	1.56 m	Rear track
$h_{cg}$	0.51 m	Height of the centre of gravity above the ground
$h_{frc}$	0.1 m	Height of the front roll centre above the ground
$h_{rrc}$	0.1 m	Height of the rear roll centre above the ground
$c_s$	1300 N s/m	Suspension damping rate
$K_{s,f}$	31.8 kN/m	Front-suspension wheel rate
$K_{s,r}$	29.7 kN/m	Rear-suspension wheel rate
$x_{lat}$	0.005 m	Lateral offset of the tyres
$\theta_c$	6°	Front caster angle
$\theta_{kpi}$	5°	Inclination angle of the kingpin
$r_r$	0.33 m	Rolling radius of the tyres
$K_{ts}$	180 kN/m	Vertical stiffness of the tyres
$C, E$	1.2, 0.2	Pacejka shape coefficients (dimensionless)
$\tau$	0.1 s	Relaxation lag time constant of the tyres
$Z_{rd}$	17.7 kN	Reference vertical load
<b>Cueing algorithm parameters</b>		
$\omega_{y1}$	9 rad/s	Lateral high-pass filter cut-off frequencies
$\omega_{y2}$	5 rad/s	
$\zeta_y$	0.8	Lateral high-pass filter damping ratio
$\omega_p$	5 rad/s	Roll high-pass filter cut-off frequency
$\zeta_p$	0.9	Roll high-pass filter damping ratio
$\omega_r$	11 rad/s	Yaw high-pass filter cut-off frequency
$\zeta_r$	0.9	Yaw high-pass filter damping ratio
$\omega_{lp1}$	9 rad/s	Tilt coordination low-pass filter cut-off frequencies
$\omega_{lp2}$	9 rad/s	
$\zeta_{lp}$	1	Tilt coordination low-pass filter damping ratio
$W_{y1}, \dots, W_{y8}$	[1 0.5 25 75 4 3 1 1]	Lateral-roll pair adaptive cost function weights
$W_{r1}, \dots, W_{r4}$	[0.5 4 3 1]	Yaw adaptive cost function weights
$P_{y0}, P_{p0}, P_{r0}$	1	Initial value of the adaptive gain
$K$	10 <sup>-4</sup>	Steepest-descent step size of the adaptive filter
$K_{oto}, \tau_{tr}, \tau_L, \tau_S$	[1 10 5 0.016]	Otolith time constants
$K_{scc}, \tau_{\sigma}, \tau_2$	[1 80 5.73]	Semicircular canal gain and time constants
$Q$	10	Linear quadratic regulator cost function weightings
$R_d$	[1 0.01 0.1 0.001]	
$R$	1	
$R$	1	
$K_y$	0.45	Body sideslip lateral scale factor
$K_p$	0.24	Body sideslip roll scale factor
$K_r$	0.35	Body sideslip yaw scale factor