

# Optimal Motion Cueing for Race Cars

Ingrid G Salisbury and David J N Limebeer (*Fellow IEEE*)

## Abstract

The application of optimal control to simulator motion cueing is examined. Existing motion cueing algorithms are hampered by the fact that they do not consider explicitly the optimal usage of simulator workspace. In this paper numerical optimal control (NOC) is used to minimise simulator platform acceleration errors, while recognising explicitly the confines of the workspace. Actuator constraints are included and the impact of restricted actuator performance is thereby facilitated. The solution of open-loop optimal control calculations are also used as a baseline against which to compare the commonly employed Linear Quadratic Gaussian (LQG) and Model Predictive Control (MPC) based techniques. The limitations of these methods are identified and two additional modules are introduced to the LQG algorithm to improve its performance.

## I. INTRODUCTION

A major challenge in dynamic driving simulator design is the reproduction of realistic vehicular accelerations within a confined workspace. The workspace of a typical simulator is of the order of metres, whereas a real car might travel several kilometres on a typical closed-circuit track. In order to bridge this gap, real accelerations are replaced by motion cues that come from filtering vehicular accelerations that are generated by high-fidelity models in order to produce platform demands that are compatible with the simulator workspace. While the simulator displacement is necessarily only a small fraction of that experienced in a real car, it can greatly improve the driver's understanding of the vehicle behaviour and the overall realism of the simulator experience.

In the design of motion cueing algorithms there are three areas that need to be addressed. The first is the motion platform; this requires a kinematic analysis and a detailed physical specification. From this a description of the platform workspace, in terms of acceleration, velocity and displacement, can be developed. The second issue relates to the optimal usage of the platform in the current race car application. Finally, once baseline cueing requirements have been established, a practical application must be developed that can run in real-time.

Current motion cueing systems use a model of the human vestibular system to determine the simulator motion that minimises the sensed acceleration error. Included in the associated optimisation cost functions are tuning parameters that can be adjusted in response to driver comments. One aim of this work is use driver feedback to characterise a priori the desirable features of race car motion cues. These are then represented mathematically in the cost function of an optimal control problem; we are not restricted to the quadratic costs employed in prior art studies. A numerical optimal control framework is then used in combination with a kinematic model of the motion platform to determine the optimal platform demands for the given driving scenario.

These results are compared with available cueing techniques with their limitations examined. Finally, modifications to the cueing algorithms are made that address the identified difficulties in order to achieve

solutions that are more nearly optimal.

In Section II an overview of a generic car simulator is given with a view to highlighting the import aspects of motion cueing. In Section III current motion cueing strategies are described and their limitations discussed. Section IV provides a description of the environment in which the motion cueing algorithms are used with the problem defined. Section V provides a kinematic model of the motion platform with an analysis of the workspace limits provided. The application of numerical optimal control to motion cueing is described in Section VI. A discussion of the workspace usage, as determined by the numerical optimal control, is provided in Section VII with these results compared with the other cueing techniques in Section VIII. Possible modifications to LQG cueing are discussed in Section IX, with the results given in Section X. The conclusions are given in Section XI.

## II. SIMULATOR COMPONENTS

A realistic driving environment is created in a high-fidelity simulator by mounting a vehicle cab on an actuated platform surrounded by audio-visual equipment. The driver cockpit is a replica of the simulated vehicle and is equipped with controls that mimic those in the car. The driver also receives realistic sensory stimuli and thus, ideally, behaves and feels as he/she would in the actual vehicle.

There are numerous systems that supply the driver with the required sensory information. Visual display graphics are of primary importance, providing images of the location and driving conditions, while also conveying the vehicle's speed. Auditory cues and haptic feedback, which is provided through the steering wheel and pedals, communicate a range of vehicle-related dynamic information to the driver including the engine speed and tyre slip. Acceleration and orientation is detected in humans by the vestibular system in the inner ear [1]. A dynamic simulator is able to stimulate this organ thereby providing an enhanced sense of the motion of the vehicle.

Figure 1 shows the integration of various subsystems into a complete simulator and illustrates how the driver generates control commands in response to sensory cues. These inputs are fed into a vehicle model that generates the vehicle-related state information including its acceleration, velocity and position. This data is then used to update the visual, audio and motion systems.

The fidelity of the simulator driving experience is necessarily limited by the quality of the cues. Advanced 3D image projection with fast refresh rates and low latency together with high-quality computer generated images (CGI) produce an excellent visual environment. In the case of motion stimuli, however, the platform workspace and bandwidth restrict the quality of the cues. This is a physical limitation rather than a technological one, since reproducing the vehicle motion exactly, requires a workspace as large as the track itself. Since this is practically infeasible, motion cueing algorithms have been developed that seek to mimic driving-related motions within the confines of the physical system.

### III. EXISTING MOTION CUEING ALGORITHMS

There are numerous motion cueing strategies that are currently applied in a variety of simulators. Before describing any new techniques, an assessment of the existing algorithms is provided, so that their limitations can be identified and exploited. Several criteria can be used to assess a cueing strategy, including

- 1) Workspace usage: how well the displacement, velocity and acceleration capabilities of the platform match those of the vehicle, and how well they are being exploited;
- 2) Tuneability: the ease with which the algorithm can be adjusted in response to driver comments;
- 3) False and missed cues: false cues refer to simulator accelerations that are opposite in sign to those produced by the car model. Missing cues refer to accelerations that are present in the car model output, but are not in the simulator motion;
- 4) Implementation: a method may in principle produce good results, but if it cannot be successfully implemented, it is of little practical value.

#### A. Classical Filtering Technique

The oldest cueing strategy, and still one of the most common, is the classical algorithm. This filtering technique was first formalised by Schmidt and Conrad in 1970 [2], in which they combined ideas from single axis systems to form the first washout circuitry for a full six degree-of-freedom (6-DOF) motion platform. This algorithm has subsequently been implemented in numerous flight and car simulators, which include both 3-DOF [3], [4] and 6-DOF platforms [5].

In this system, simulator motion demands are generated by high-pass filtering the translational acceleration and angular velocity signals produced by the car model. The low frequency (longer duration) components of acceleration signals result in large displacements, which cannot be replicated in a simulator. Through an iterative, trial-and-error tuning process the filter gains and cut-off frequencies are adjusted so that the platform position demand remains within the confines of the workspace.

The lack of low-frequency acceleration information in the longitudinal and lateral directions can be partially mitigated by the addition of tilt-coordination. This technique exploits the fact that humans cannot distinguish between a steady-state tilt and a translational acceleration; in order to simulate a longitudinal or lateral acceleration the driver is pitched or rolled. Providing the visuals are moved appropriately, and the rotational motion is sufficiently slow, the driver will perceive a steady-state forward or sideways acceleration and not a tilt. A rate-limit on the platform rotation produces a delayed peak in the sensed acceleration. Therefore, if the acceleration changes direction rapidly, the platform moves too slowly and this technique is not usable.

As illustrated in Figure 2, the pitch and longitudinal (and lateral and roll) freedoms are considered in combination. The translational acceleration is high-pass filtered in order to generate a position reference for the platform. It is also low-pass filtered and added to the rotation channel in order to produce tilt coordination in the rotation demand. The translational acceleration may be scaled and limited so that it is not interpreted as a

rotational motion. An additional platform-centering washout filter must be included in each channel. There is no cross-coupling between the yaw and heave freedoms.

A well-documented disadvantage of the classical approach is the generation of false cues that occur after an onset acceleration; this is a feature of any linear time-invariant filter. As explained in [6], this backlash effect is a characteristic of linear high-pass filters, and is an intrinsic limitation of any linear technique. A method is introduced in [6] that reduces the miscues through the addition of a non-linear post-filtering module. An adaptive gain is applied to the output of the filter, which reduces the amplitude of the signal during miscuing.

Another limitation of this approach is that it does not explicitly account for the platform workspace limits [5]. As a result, the filters are either tuned for the anticipated worst-case acceleration, or for a particular driving session. As noted in [7] the filters are usually tuned to accommodate the maximum amplitude acceleration and so smaller onset cues are not sustained for as long as they should be.

Despite the limitations of this technique it is easy to implement and simple to adjust in response to driver comments [8], hence its enduring popularity.

#### *B. Non-Linear Coordinated Adaptive Algorithm*

The classical algorithm is tuned to accommodate worst case accelerations and so poor performance relative to low-amplitude cues results. Adaptive techniques aim to overcome this limitation by varying the cueing filter parameters in real-time. These (adaptive) algorithms can be designed to adjust the translational high-pass, cross-term and rotational filters, or any combination of these. The filter gains and cut-off frequencies are calculated at each time step to minimise a cost function of the form [5]

$$J = (a_{ref} - a_{sim})^2 + w_1 v_{sim}^2 + w_2 p_{sim}^2 \quad (1)$$

in which  $a_{ref}$  is the acceleration of the vehicle, and  $a_{sim}$ ,  $v_{sim}$  and  $p_{sim}$  are, respectively, the acceleration, velocity and displacement of the platform, while  $w_1$  and  $w_2$  are weights. The cost is a weighted sum of the platform position, velocity and the acceleration error; this error is the difference between the input and output of a dynamic system. A steepest descent method is used to find the cost function minimum [9]. By adjusting the weights in the cost function the algorithm can be tuned so that the resulting acceleration demand lies within the platform workspace.

This method shows an improvement in the fidelity of the motion of the simulator platform when it is close to the neutral position, and accelerations are only corrupted as the platform nears its limits. The cost function can be defined in different ways making the method more flexible and intelligent [8]. The tuning is also more intuitive as the cost function weights cause a clear trade-off between the fidelity of the acceleration and the extent of the platform motion [9]. The experimental results in [9] and [10] show a reduction in the false cues produced by this algorithm as compared with the classical technique.

The disadvantages of this method are that it is computationally demanding, and so has a higher execution

time, with the algorithm being difficult to adjust [8]. It has also been found that for certain motion trajectories the adaptive filter distorts the motion perception [11]. These authors suggest a more complex cost function that considers lateral and roll freedoms in combination. A model of the vestibular system is included and the adaptive cost function is adjusted to minimise either the rotational sensory error or the roll angle. This showed an improved performance in the angular sensations, but the translational perception was compromised. The cost function can also be adjusted to penalise acceleration errors when the derivative of the acceleration is high [11], thereby including a commonly employed principle of motion cueing — cue for sudden changes in motion.

### C. Optimal Cueing

This cueing approach uses linear quadratic regulator theory to design optimal washout filters; a detailed description of the algorithm can be found in [8], [12] and [13] and later in this paper.

The aim of this approach is to design a filter  $F(s)$  that relates the simulator acceleration demands to the vehicle reference accelerations as shown [10]:

$$u_{sim}(s) = F(s)u_{ref}(s). \quad (2)$$

This filter is designed using LQR theory to minimise a cost function of the form:

$$J = \int_0^\infty (e^T Q e + x_{sim}^T R_d x_{sim} + u_{sim}^T R_s u_{sim}) dt \quad (3)$$

Humans sense translational accelerations and angular velocities using the vestibular system in the inner ear. The mechanics of sensation introduce filtering that can be characterised as transfer functions from the ‘real’ to sensed, or perceived acceleration [14]. By incorporating this filtering process in the plant dynamics, the perception error,  $e$  in (3) can be determined as the difference between the acceleration sensed by the vehicle and simulator drivers.

The variables  $x_{sim}$  and  $u_{sim}$  represent the simulator states and input accelerations respectively. The weights  $Q$ ,  $R_d$  and  $R_s$  are tuned iteratively so that the platform stays within its limits for the worst-case acceleration input.

The filter structure that results from this technique is similar to the classical algorithm. By using a model of the vestibular system (where the specific force in the translational direction is a function of the platform’s rotational motion) the longitudinal, pitch and cross-term filters are designed simultaneously. These filters are usually 3rd to 5th order [8].

The tuning of this algorithm is transparent to the user as it only involves adjusting the weights that trade-off platform workspace usage and vestibular error [9]. This approach incorporates directly a model of the vestibular system and so takes advantages of the natural filtering of humans. The disadvantages of this method is that the freedoms are assumed to be decoupled and the vehicle motion a stochastic signal [8], which are not realistic in a practical simulator. It also shares the limitations of classical filters in that the system is tuned for worst case

accelerations, and so does not use the full workspace in normal operation, and it suffers from false cueing.

#### *D. Prepositioning*

Prepositioning is not a motion cueing algorithm in itself, but it is a useful technique that can be employed in conjunction with the classical and optimal approaches in order to improve workspace usage. Developed by Weiß [15], the idea behind this technique is to anticipate the next movement of the vehicle and move the platform to a point that will increase the workspace in the required direction. If the driver is going to brake (accelerate), the platform should move to the foremost (rearmost) longitudinal limit of the workspace. In this way the workspace for braking, accelerating and cornering is increased and optimal or classical cueing algorithms can be tuned to give stronger or longer duration cues. The crux of this technique is moving the platform sufficiently slowly that the pre-maneuvring is not detected by the driver and does not interfere with the other motion cues or introduce new ones.

Weiß presents two techniques to implement prepositioning [15]. The first is through filtering a neutral-position-offset demand to ensure a smooth transition of the platform to the prepositioning location. The filtering technique allows a slow onset and final motion, however, a limitation lies in the maximum velocity that the platform needs to achieve in order to move within the prediction time. The second technique is a logic method. The desired position and prediction time are used to calculate the velocity at which the platform must move to get there. At first this was aimed to be at sub-detectable velocities, but this does not ensure that the platform reaches the position in time and was found to be inadequate. The filtering technique is smoother, but the logic technique requires a lower peak velocity.

The limitation of the prepositioning technique is that it requires the ability to predict the future vehicle motion. In the longitudinal direction it is possible to estimate the upcoming accelerations by observing the current vehicle speed or gear ratio (braking is likely to follow high speed driving, while accelerating follows a period of low speed) [16]. The lateral direction can be prepositioned from the track data. The curve radius, vehicle speed and an estimated time-to-next-corner can be used to move the platform into a good position [16]. Finally, [16] use non-linear rate limiters to keep prepositioning movements below the driver perception thresholds.

#### *E. MPC*

Recent motion cueing algorithms have explored the use of model predictive control [17]. These methods also minimise the perception error, but the platform limits and the driver motion perception thresholds are included as hard constraints. At each time step the signals are calculated over a prediction horizon  $N$ , where the current control action is the initial condition [17].

The result is that the simulator follows the reference acceleration signal until it nears the platform limits, at which point it slows the platform down to a level imperceptible to the driver. This yields smooth decelerations

near the limits. Washout to the neutral position is not included, but is implemented as a parallel process that is activated by the accelerations and driver actions.

Testing with this method has demonstrated better workspace usage than the classical technique [17]. It is also the simplest of the cueing algorithms to tune [9]. However, in the period where the platform slows towards the limits, none of the high-frequency content of the motion is present [9].

#### IV. PROBLEM DESCRIPTION

##### A. Race Car Applications

While there are common criteria for all driving simulators, there are also situation-specific differences that must be addressed. The simulator discussed here is used in race car applications, where the primary functions are driver training and car set-up ahead of races.

Race car driving is characterised by high levels of longitudinal and lateral acceleration (in excess of 4Gs), which cannot be sustained in a confined simulator workspace. This makes the motion cueing problem challenging and it is important to identify the information the drivers wish to receive from the motion platform. Discussions with the drivers make it clear that they appreciate that a simulator cannot mimic a real race car. However, they require good feedback about the vehicle behaviour in certain scenarios:

- 1) When the brakes are applied, a strong onset cue is required that communicates the car's rate of deceleration;
- 2) During the later part of the braking phase the driver needs to know if any of the wheels are 'locking up' so that braking can be correspondingly regulated;
- 3) During cornering the driver needs to know whether the car is under, or over steering.

Understanding the cues that the driver's require assist in both the design and assessment of cueing algorithms.

A further characteristic of this application is that the car is typically driving around a known closed track. In racing the aim is to minimise lap-time, and to achieve this there is an optimal speed profile and racing line [18, 19]. Given that experienced drivers generally drive the car close to optimal, there is little variation in the acceleration profiles between laps. As a result the braking and cornering situations are highly predictable.

##### B. Cueing problem

This section provides a detailed analysis of the acceleration and angular velocity trajectories that need to be addressed. Table I contains a representative range of the accelerations, velocities and displacements a race car will experience during a lap of a track. The two graphs in Figure 3 display the frequency content of the acceleration signals for a typical lap. There are some important problem features that are highlighted in this data. Firstly, in the translational freedoms, both longitudinal and lateral motions contain more low-frequency information as compared to the heave direction. The vertical displacement is small and it should be possible to reproduce this exactly in a simulator setting. This result is intuitive when one thinks of the motion of a vehicle on a hard suspension system. Similarly, in the rotational freedoms, the roll and pitch angles are only a

TABLE I  
TABLE OF TYPICAL RANGES IN ACCELERATION, VELOCITY AND DISPLACEMENT IN THE VEHICLE 6-DOF; ALL QUANTITIES ARE IN SI UNITS WITH THE EXCEPTION OF ANGLES THAT ARE GIVEN IN DEGREES.

Direction	Acceleration	Velocity	Displacement	Frequency
Longitudinal	-40:20	-70:70	-650:500	0-5
Lateral	-50:50	-65:72	-800:200	0-5
Heave	-15:15	-0.5:0.5	-0.3:0.25	2-7
Yaw	-20:20	-1.2:1.1	0:360	0-12
Pitch	-30:20	-0.65:0.6	-2:0.5	2-8
Roll	-65:80	-0.8:1.2	-2:5	2-25

few degrees. These angular freedoms should also require little, or no filtering before being reproduced by the simulator. The outcome of this analysis is a reduction of the motion cueing problem from designing cueing for 6-DOF to only 3-DOF (longitudinal, lateral and yaw).

### C. Vehicle Model Data

The vehicle response data for this study is generated by the minimum-lap-time optimal control simulation described in [18, 19]. This 3DOF car model includes tyre and aerodynamic forces, and captures the vehicle's low-frequency drive dynamics, but not the higher frequency car ride (suspension) dynamics. Since workspace analysis is concerned with the filtering of low-frequency data, which results in large displacements, the use of a simplified vehicle model in this context is legitimate. The higher frequency ride dynamics that come from the vehicle suspension can be rendered by the simulator without filtering as a separate exercise and are not a necessary component of this analysis.

## V. HEXAPOD MODEL

The motion platform used in this study is a conventional Stewart Platform. The cab, which consists of the race car cock-pit, screens and projectors, is mounted on six actuated legs, which are connected to the platform with spherical joints and to the ground-mounted base plate with pin joints. By controlling the six actuated leg lengths the cab can be moved with 6-DOF; see Figure 4.

### A. Inverse Kinematics

Since the boundaries of the hexapod workspace are a direct consequence of the actuator stroke limits (maximum and minimum leg lengths) it is necessary to determine the hexapod leg lengths for a given platform position. These calculations are referred to as the inverse kinematics and are used in the design of motion cueing algorithms to ensure the limits of the platform are observed. The development of the kinematics equations is a standard procedure and can be found in [20].

1) *Actuator Length:* Referring to Figure 4, an inertial reference frame  $xyz$  with origin  $O$  is defined at the geometric centre of the hexapod base. A second body-fixed frame  $x'y'z'$  is similarly defined with its origin  $C$



at the geometric centre of the platform. The translation of the platform from the origin of the inertial reference frame is described by the vector  $\underline{S}$  and the matrix  $R$  describes the platform orientation in terms of 3 Euler angles. For consistency with the vehicle modelling, the 3-2-1 Euler angle sequence is adopted. This corresponds to a first rotation about the  $z$ -axis (yaw angle ( $\psi$ )), then about the  $y$ -axis (pitch angle ( $\theta$ )) and finally about  $x$ -axis (roll angle ( $\phi$ )). The rotation matrix is given

$$R = R_z(\psi)R_y(\theta)R_x(\phi) \quad (4)$$

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \quad (5)$$

where  $\cos(\theta)$  is abbreviated as  $c_\theta$  and  $\sin(\theta)$  as  $s_\theta$ , and so on.

The base joints of the hexapod legs are described by six vectors  $\underline{B}_j$  in the inertial frame, and the platform joints are described by vectors  $\underline{p}_j$  in the body-fixed frame. The platform joints can then be described by vectors  $\underline{P}_j$  in the inertial frame, which are related to  $\underline{p}_j$  by:

$$\underline{P}_j = \underline{S} + R \cdot \underline{p}_j \quad (j = 1, \dots, 6) \quad (6)$$

The platform legs can be described by vectors  $\underline{l}_j$ , which point from the base joints to the corresponding platform joints and can be found as the difference between the platform and base vectors. The leg lengths ( $l_j$ ) are simply the magnitude of the leg vectors and  $\hat{\underline{l}}_j$  are unit vectors associated with the directions of the legs.

$$\underline{l}_j = l_j \hat{\underline{l}}_j = \underline{P}_j - \underline{B}_j \quad (j = 1, \dots, 6) \quad (7)$$

2) *Actuator Velocity:* Suppose that the leg angular velocities are  $\underline{\omega}_{Lj}$  and that their vector-valued rates of change of length  $\dot{l}_j$  have magnitudes  $\dot{l}_j$  in directions  $\hat{\underline{l}}_j$ . Thus

$$\dot{\underline{P}}_j = \dot{l}_j \hat{\underline{l}}_j + \underline{\omega}_{Lj} \times \underline{l}_j. \quad (8)$$

It follows from (6) that

$$\dot{\underline{P}}_j = \dot{\underline{S}} + \underline{\omega} \times \underline{p}_j. \quad (9)$$

Equating (8) and (9) and dotting the result with  $\hat{\underline{l}}_j$  gives

$$\dot{l}_j = \begin{bmatrix} \hat{\underline{l}}_j & \underline{p}_j \times \hat{\underline{l}}_j \end{bmatrix} \cdot \begin{bmatrix} \dot{\underline{S}} \\ \underline{\omega} \end{bmatrix} \quad (10)$$

since  $\underline{l}_j$  is parallel to  $\hat{\underline{l}}_j$ .

The leg angular velocities can be found from (8) as follows

$$\hat{l}_j \times \dot{\underline{p}}_j = \hat{l}_j \times (\dot{l}_j \hat{l}_j + \underline{\omega}_{Lj} \times \underline{l}_j) \quad (11)$$

$$= \hat{l}_j \times \underline{\omega}_{Lj} \times \underline{l}_j \quad (12)$$

$$= l_j \underline{\omega}_{Lj}, \quad (13)$$

since  $\underline{\omega}_{Lj}$  is perpendicular to  $\hat{l}_j$ . Whence  $\underline{\omega}_{Lj} = (\hat{l}_j \times \dot{\underline{p}}_j)/l_j$ .

3) *Actuator Acceleration:* The accelerations of the platform joints can be found by differentiating (10) as follows

$$\ddot{l}_j = (\ddot{\underline{S}} + \underline{\dot{\omega}} \times \underline{p}_j) \cdot \hat{l}_j + (\underline{\dot{S}} + \underline{\omega} \times \underline{p}_j) \cdot \dot{\hat{l}}_j \quad (14)$$

since  $\dot{\underline{p}}_j = 0$ . Whence

$$\ddot{l}_j = \begin{bmatrix} \hat{l}_j & \underline{p}_j \times \hat{l}_j \end{bmatrix} \cdot \begin{bmatrix} \ddot{\underline{S}} \\ \underline{\dot{\omega}} \end{bmatrix} + \frac{d}{dt} \left( \begin{bmatrix} \hat{l}_j & \underline{p}_j \times \hat{l}_j \end{bmatrix} \right) \cdot \begin{bmatrix} \underline{\dot{S}} \\ \underline{\omega} \end{bmatrix} \quad (15)$$

in which we can use (7) to show that

$$\dot{\hat{l}}_j = (\dot{\underline{p}}_j - \dot{l}_j \hat{l}_j)/l_j. \quad (16)$$

## B. Hexapod Workspace

An important feature of the hexapod motion platform is the shape of the workspace. Since movement in 6-DOF is created by the same six actuators, motion in one direction affects the capabilities of the platform to move in another. In this application large displacements are required in the longitudinal, lateral and yaw directions, and so the workspace usage relationships between these 3 freedoms are examined. Figure 5 shows the workspace limits for the lateral and longitudinal motion for various values of platform yaw angle. By focussing temporarily on the heavy (black) plot, which corresponds to a yaw angle of zero, one observes that movement in either the longitudinal or lateral direction reduces the workspace available in the other. The leg configuration employed in this study makes the zero-yaw-angle workspace symmetrical about the x-axis. The workspace perimeter is reached when one or more of the leg actuators reaches its maximum or minimum length limit. The six vertices in this figure correspond to points in configuration space where multiple legs saturate and there is a change over between saturating legs.

Figure 5 also shows the lateral and longitudinal workspace bounds for increased values of yaw angle. As the platform rotates (yaws), the workspace is no longer symmetrical and it becomes smaller and more ‘triangular’ in shape. This irregularly shaped workspace has consequences for the motion cueing algorithm design. In most cueing algorithms each freedom is tuned individually for a predetermined workspace. If the actual available workspace is less than expected, because of workspace coupling, the platform may not fully render the cueing signal. Consequently, it may be better to design for worst-case displacement limits, even though this results in

the workspace not being fully utilised.

## VI. NUMERICAL OPTIMAL CONTROL

Current cueing algorithms produce a cueing signal that is a filtered and/or attenuated version of the reference with the filter parameters adjusted until the platform remains within the workspace. None of these strategies directly address the problem of seeking best usage of the available workspace. The LQG method, for example, constrains the output to be a linearly filtered version of the input, and does not attempt to make optimal use of the workspace. The MPC method recognises constraints on the workspace, but it results in a scheme that cues the driver accurately until the platform limits are reached, and then produces miscues.

In a racing situation the acceleration profile for a lap of a given track is essentially known, although there may be some minor variations between drivers. The aim of this study is to determine the optimal way to use the platform workspace in order to cue the driver over an entire lap. Numerical optimal control is applied to this problem with the details described next.

### A. Optimal Control

The purpose of an optimal control calculation is to determine the state and control associated with a system in order to minimise a performance index [21]. When expressed in Bolza form, the performance index is given by

$$J = \Phi(t_0, x(t_0), t_f, x(t_f), p) + \int_{t_0}^{t_f} l(t, x(t), u(t), p) dt, \quad (17)$$

while the system and operating constraints are given by

$$\left\{ \begin{array}{ll} \frac{dx}{dt} - f(t, x(t), u(t), p) &= 0 \\ g(t, x(t), u(t), p) &= 0 \\ h(t, x(t), u(t), p) &\leq 0 \\ g_b(x(t_0), x(t_f), u(t_0), u(t_f), p) &= 0 \end{array} \right. \quad (18)$$

where  $t_0 \leq t \leq t_f$  is the optimisation interval with  $t_f$  either fixed, or free to be optimised. The vector  $p \in \mathbb{R}^{n_p}$  contains any fixed parameters to be optimised<sup>1</sup>;  $x(t) \in \mathbb{R}^n$  and  $u(t) \in \mathbb{R}^m$  are the state and control vectors respectively. The vector-valued function  $f(\cdot) \in \mathbb{R}^n$  describes the system dynamics. The vector functions  $g(\cdot) \in \mathbb{R}^{n_g}$  and  $h(\cdot) \in \mathbb{R}^{n_h}$  define the equality and inequality constraints for the system. The subscript  $b$  refers to the boundary constraints with  $g_b(\cdot) \in \mathbb{R}^{n_{g_b}}$ . The scalar function  $l(\cdot)$  is the stage cost that is a function of the state, the controls and the parameters.

Direct methods of the type employed here transcribe, or convert, infinite-dimensional optimal control problems into a finite-dimensional optimisation problem with algebraic constraints; a nonlinear programming problem (NLP) [21].

<sup>1</sup> $\mathbb{R}^n$  denotes the set of  $n$ -dimensional real vectors.

The work in this paper makes use of a Gauss-Legendre-Radua psuedospectral optimal control solution method that has been implemented in the software package GPOPS-II [22, 23].

### B. Optimal Cueing Problem Formulation

In an ideal situation the aim of a simulator motion cueing algorithm is to reproduce the accelerations a driver would experience in a real car. If the car's translational accelerations are of the form  $a_{ref}^x(t)$  (where the superscript defines the direction of motion), then the acceleration sensed by the driver is given by  $\hat{a}_{ref}^x(s) = V(s)a_{ref}^x(s)$ , in which  $V(s)$  is a model of the human otoliths that are part of the human vestibular system [24];  $s$  is the Laplace variable. The aim of a cueing algorithm is to make the difference  $\hat{a}_{ref}^x(t) - \hat{a}_{sim}^x(t)$  as small as possible. In the yaw freedom the sensed yaw velocity is given by  $\dot{\psi}_{ref}(s) = W(s)\dot{\psi}_{ref}(s)$ , where  $W(s)$  is a model of the semi-circular canals. The difference  $\dot{\psi}_{ref}(t) - \dot{\psi}_{sim}(t)$  must be minimised. In the simulator, the motion cues are generated by a hexapod, which has limited displacement, velocity and acceleration capabilities; these limitations prevent the simulator from matching exactly the motions of the vehicle.

The aims and constraints described above can be embedded in an optimal control problem with a quadratic form type of performance index

$$J = \int_{t_0}^{t_f} \underline{y}(t)^T Q \underline{y}(t) dt. \quad (19)$$

The optimal control problem set up is shown in Figure 6.

The problem constraints are of the form

$$\dot{\underline{x}}(t) = f(\underline{x}(t), \underline{u}(t)) \quad (20)$$

$$\underline{y}(t) = g(\underline{x}(t), \underline{u}(t)) \quad (21)$$

$$\underline{l}_{min} \leq \underline{l}_{act}(t) \leq \underline{l}_{max} \quad (22)$$

$$\underline{v}_{min} \leq \underline{v}_{act}(t) \leq \underline{v}_{max} \quad (23)$$

$$\underline{a}_{min} \leq \underline{a}_{act}(t) \leq \underline{a}_{max} \quad (24)$$

in which  $f(\underline{x}(t), \underline{u}(t))$  describes the plant and vestibular model dynamics. The remaining constraints are associated with the motion platform. The optimal control problem is to determine the input  $\underline{u}(t)$  such that the performance index is minimised with the constraints respected.

As shown in Figure 6, the control input is a vector of simulator longitudinal and lateral accelerations and yaw velocities  $\underline{u}(t) = [a_{sim}^x(t), a_{sim}^y(t), \dot{\psi}_{sim}(t)]^T$ . This input and the motion demand  $[a_{ref}^x(t), a_{ref}^y(t), \dot{\psi}_{ref}(t)]^T$  are used to compute the system states  $\underline{x}(t)$  and the outputs  $\underline{y}(t)$  from the system dynamics described in (20) and (21). The system dynamics relate the reference accelerations and the control input, through a nonlinear hexapod model and linear vestibular system model, to the actuator accelerations, velocities and displacements

as well as the system output  $\underline{y}(t)$ . The actuator accelerations, velocities and displacements are components of the state vector and are limited in (22), (23) and (24) respectively. The output  $\underline{y}(t)$ , which contains the error (the difference between the motion sensed in the simulator and the car) is a function of the system state.

$$\underline{y}(t) = \begin{bmatrix} \hat{a}_{ref}^x(t) - \hat{a}_{sim}^x(t) \\ \hat{a}_{ref}^y(t) - \hat{a}_{sim}^y(t) \\ \dot{\hat{\psi}}_{ref}(t) - \dot{\hat{\psi}}_{sim}(t) \end{bmatrix} \quad (25)$$

In sum, the cost function must be selected to minimise the weighted sensed translational acceleration error and the weighted sense rotational velocity error. This is an iterative process that involves a lot of driver testing, who expect the simulator to provide the cues that are important to high-performance driving.

### C. Cost Function Assessment and Development

In the longitudinal direction the braking and accelerating scenarios need to be cued, with the numerical optimal control used to determine the optimal acceleration profile for each situation.

Figure 7 shows a braking manoeuvre (black curve) with the simulator trajectory (dashed red curve) computed to minimise the square of the acceleration error; a standard time-invariant quadratic cost of the form (19) is used. The optimised response is constrained to begin the manoeuvre with the platform at the front of the workspace and finish it at the rear. In this way the platform was thus ‘prepared’ for braking with the manoeuvre ended in a position that is ready to accelerate. There are two features of this cueing signal that deserve comment. The first is the delayed response, which occurs so that the peak deceleration of the platform coincides with that of the car. The second feature of the motion is the miscue that brings the platform to a halt towards the end of braking manoeuvre; this is a consequence of poor workspace management. This response is not satisfactory, because it does not provide the strong initial onset cue required by the race car drivers we consulted. From discussions it was established that drivers require a strong initial onset cue that is then followed by a miscue of minimum possible magnitude. The cueing acceleration resulting from (19) does not exhibit these characteristics and thus required modification.

The following cost function is an adjusted version of (19) that is used to induce more desirable response characteristics

$$J = \int_{t_0}^{t_f} (\hat{a}_{error}^x)^2(t)Q_1w(t) + (\hat{a}_{sim}^x)^2(t)Q_2(1 - w(t))dt \quad (26)$$

where

$$\hat{a}_{error}^x(t) = \hat{a}_{ref}^x(t) - \hat{a}_{sim}^x(t) \quad (27)$$

and

$$w(t) = \frac{1}{2}(1 - \tanh(\beta(t - t_s))). \quad (28)$$

The first term in the cost function penalises the acceleration error during the first part of the braking phase.

The second term then penalises the platform acceleration during the remainder of the braking manoeuvre. The weights  $Q_1$  and  $Q_2$  are used to balance the relative strengths of the onset cue and the miscue; not surprisingly, the stronger the onset cue, the worse the miscue. The function  $w(t)$  is a ‘smooth’ switch that is 1 from  $t = 0$  to  $t = t_s$  and 0 thereafter; the constant  $\beta$  controls the sharpness of the switch. The switching time and weights are selected to produce the strongest, and longest onset cue that can be achieved within the actuator constraints. Since braking manoeuvres have similar strengths, the tuning parameters are consistent for a given lap. The result of this adjusted cost function is shown as the (dashed blue curve) in Figure 7. The acceleration cue contains a strong timely onset cue, while the false cue has been extended, but with a much reduced amplitude.

The accelerating scenario is examined next. The cost function used in this case is

$$J = \int_{t_0}^{t_f} (\hat{a}_{error}^x)^2(t) dt \quad (29)$$

and in this case the platform is constrained to start at the back of the workspace and finish at the front of it. The result over a typical accelerating manoeuvre is shown in Figure 8. The cueing acceleration retains the general shape of the acceleration reference, but the low-frequency information has been removed. Unlike the braking case, there are no discernible onset cues and so the general approach is to cue acceleration changes and a simple quadratic cost function of the form (19) is suitable in this case.

Lateral acceleration cues have a very different shape (as compared with the longitudinal case) and may be of high amplitude and long duration in a protracted high-speed corner. This type of acceleration cannot be replicated in a simulator with a confined work space. Again the onset acceleration is cued, but not sustained. For these reasons the onset cue generated by (26) is used again; cost of the form (29) proved unhelpful. The lateral cue for a cornering manoeuvre is shown in Figure 9.

The yaw acceleration has a similar shape to the lateral cue and so a cost function of the form (26) is used in this case too.

## VII. RACING LAP OPTIMAL CUEING

The cost functions previously developed are now used to determine an optimal simulator cueing strategy for a lap of the Barcelona Circuit de Catalunya. The results focus on corners 1-4, shown in Figure 10. This section of the track is chosen because it features high-amplitude braking cues, high-speed corners (3 and 4) and two consecutive right-hand corners (the relevance of which is discussed in Section VIII on lateral acceleration cueing). In Sections VII-A, VII-B and VII-C, the longitudinal, lateral and yaw cueing demands are computed separately. In Sections VII-D the longitudinal and lateral cues are considered in combination, with three-dimensional cueing assessed in Sections VII-E. Multi-dimensional cueing is important, because the various degrees of freedom are coupled, and motions in one dimension might compromise those in another.

### *A. Longitudinal Acceleration Cueing*

The longitudinal acceleration plot is given in Figure 11. The purpose of the open-loop numerical optimal control solution is to provide insight into the best use of the platform workspace. As can be seen, the braking manoeuvres are characterised by strong onset cues, while during acceleration changes are tracked, although somewhat attenuated. Under braking, the motion cueing is being restricted by the velocity capability of the platform. By examining the velocity and displacement signals it is seen that the velocity achieves its peak value before the platform reaches the displacement limit. It is thus the velocity limit that prevents the onset cue from being sustained and prevents the platform workspace from being fully exploited. The platform is prepositioned slightly forwards, or slightly rearwards, of the neutral position in anticipation of the next manoeuvre. Since the workspace is not being fully utilised, there is no need for extreme positioning prior to the next manoeuvre.

### *B. Lateral Acceleration Cueing*

Unlike cueing in the longitudinal direction, the lateral acceleration trajectory can have two consecutive manoeuvres in the same direction; see Figure 12 between 10 and 15 s, and between 17 and 23 s. In this case it would be best to position the platform away from the same extreme edge of the workspace at the start of each manoeuvre. However, between corners, the platform will then need to move in the opposite direction to the vehicle acceleration and a false cue will result. A balance therefore needs to be established between a miscue and the workspace needed for the onset cue. Since the full workspace is not being utilised, because of velocity saturation, the platform need not be prepositioned at an extreme in the workspace in this case.

### *C. Yaw Velocity Cueing*

The yawing motion differs from the longitudinal and lateral accelerations in that the cues can be sustained for longer and in certain circumstances almost fully reproduced by the platform, if its position limits are not exceeded. As with the lateral freedom, there can be multiple manoeuvres in the same direction. In this case, a bad miscue is required to preposition the platform, this can be seen from 12-17 seconds in Figure 13. Despite this the platform does not start from the workspace extreme - there is a trade-off between the miscue required to move there and the improvement the additional workspace has on the actual cue.

### *D. Longitudinal and Lateral Combined Cues*

Having considered each DOF separately, the next step is to combine the two translational freedoms. Figure 14 contains the lateral and longitudinal cueing signals produced by the optimal control solver. The most noticeable difference between the responses in this case, and those achieved previously, are the stronger miscues, although the braking and cornering onset cues maintain a similar magnitude and duration. The source of this change arises, once again, from the platform velocity limits. By examining the platform behaviour during the first braking manoeuvre, one sees that the longitudinal deceleration is cued with the platform then reaching its

maximum backward velocity. Two of the platform actuators are therefore moving at their maximum speed and so the platform does not have the ability to move laterally, as this would exceed the velocity limits of these legs. Therefore, in order to cue the lateral onset, the longitudinal velocity needs to be reduced and a miscue is generated in consequence.

From the displacement signals given in Figure 15 it is observed that the platform does not reach the maximum longitudinal or lateral displacement, because at these extremes, no movement would be possible in the other dimension.

By examining the xy-displacement of the platform in Figure 16, it can be seen that while the platform reaches the boundaries, the workspace is not fully utilised during this 4-corner manoeuvre due to the actuator velocity limits. Indeed, as a result of these velocity constraints, optimal cueing can be achieved in a smaller workspace. The blue trajectory in Figure 16 is the xy-displacement of the platform in the case that the actuator velocity limits are doubled. In this case the platform makes better use of the available workspace, since the trajectory has longer subarcs and the platform travels further in both the longitudinal and lateral directions.

#### E. Full 3-DOF Cueing Simulation

Adding the yawing freedom to the simulator's kinematics has important implications for the utilisation of the workspace. Figure 5 shows that the motion platform's translational workspace is reduced significantly by simultaneous yawing. For that reason it is necessary to assign a weighting to each of the various freedoms so that a balanced performance is achieved:

$$J = \int_{t_0}^{t_f} (J_x + J_y + W J_\psi) dt \quad (30)$$

where:

$$J_x = (\hat{a}_{error}^x)^2(t) Q_1^x w^x(t) + (\hat{a}_{sim}^x)^2(t) Q_2^x (1 - w^x(t)) \quad (31)$$

$$J_y = (\hat{a}_{error}^y)^2(t) Q_1^y w^y(t) + (\hat{a}_{sim}^y)^2(t) Q_2^y (1 - w^y(t)) \quad (32)$$

$$J_\psi = (\hat{\psi}_{error})^2(t) Q_1^\psi w^\psi(t) + (\hat{\psi}_{sim})^2(t) Q_2^\psi (1 - w^\psi(t)). \quad (33)$$

In index (30), the cost in each freedom is based on the performance index of equation (26). The total cost is a weighted sum of the cost in each freedom. The longitudinal and lateral freedoms are weighted equally, while the yaw freedom is assigned a weight  $W$ . Since yaw rotation rapidly degrades the translational capabilities of the platform it is given relatively lower importance ( $W \ll 1$ ).

Figure 17 shows the translational acceleration and rotational velocity trajectories, which result from giving the yaw angle low weighting. The translational freedoms retain their prior performance, but the yaw angle is only cued for a short-duration onset.

Figures 18 shows the contrasting performance resulting from a higher yaw angle weight in (30). In this case



the yaw angle is cued accurately, but this results in a degraded response and an increase in the miscues in the translational freedoms. This analysis illustrates a limitation of the hexapod as a high-performance motor vehicle motion platform and provides a means with which to balance the trade-off between rotational and translational motion.

### VIII. COMPARISON OF CUEING TECHNIQUES

The numerical optimal control results are now compared with established LQG and MPC techniques. These methods are only implemented for the longitudinal freedom, with the results of a typical braking manoeuvre analysed.

#### A. Linear Quadratic Gaussian

In this application cueing filters are required for the longitudinal, lateral and yaw freedoms, but only the longitudinal freedom under braking will be discussed here. The structure of the motion cueing problem for a single DOF is shown in Figure 19. The acceleration produced by the car model,  $a_{ref}^x$ , is the system input. The top signal path generates the vestibular response of a driver in the “real” car. The vehicular acceleration is filtered by the vestibular system to generate a perceived motion sensation,  $\hat{a}_{ref}^x$ . The lower signal path represents the simulator, where the vehicle accelerations are first filtered by  $F(s)$  to produce acceleration demands,  $a_{sim}^x$ . The driver’s vestibular system filters the platform motion to produce a perceived motion,  $\hat{a}_{sim}^x$ . The perception error is defined as the difference between the motion sensed by the driver in the car and by the driver in the simulator;  $\hat{a}_{ref}^x - \hat{a}_{sim}^x$ . This error is frequency shaped by  $w_1(s)$ , and the result  $z_1$  forms the first output. The remaining three outputs  $z_2, z_3, z_4$  are frequency weighted simulator acceleration, velocity and displacement signals respectively. The inclusion of frequency weighting is an extension of the standard LQG approach reported elsewhere, and allows different frequency-response aspects of the signal to be emphasised. The filter design problem can be solved by embedding it in the standard generalised regulator shown on the right of Figure 19 [25]. The filter  $F(s)$  is synthesized so as to minimise the  $\mathcal{H}_2$ -norm of the transfer function  $a_{ref}^x \mapsto z$  in which  $z = [z_1 \ z_2 \ z_3 \ z_4]'$ . This cost function clearly results in a trade-off between the perception error and the simulator motion. If the platform motion is heavily penalised, a larger perception error will result, conversely if the perception error is to be small increased platform motion is required. With these simple principles in mind, the weights can be tuned until the platform remains within its workspace constraints for the desired test scenario.

Traditionally the LQG-derived filter is adjusted in response to driver comments. Changing the weights in the cost function not only modifies the simulator workspace usage, while still minimising the perception error, but also changes the fine detail of the acceleration cue. In this application, the NOC produced acceleration has been adjusted to contain the desired cueing features. Using the NOC produced accelerations as a target response, the LQG-derived filter is tuned to approximate this as closely as possible.

Once again during the onset cue the velocity approaches the constraint; see the solid green curve in the lower figure. The platform does not use the work space effectively, because the LQG method is difficult to tune in order to fully exploit the workspace without unacceptable levels of actuator saturation. As with prior braking studies, the platform was pre-positioned near the front of the workspace, but it did not come back past the neutral position over the entire manoeuvre.

The primary difference between the LQG and numerical optimal control approaches lies in the shape, and timing of the miscue. The LQG filter produces a weaker onset cue that is combined with a miscue in the terminal stages of braking; this miscue has a higher amplitude than that produce by the optimal control strategy. The shape and duration of this false cue are a direct result of using a linear washout filter. This miscue cannot be changed without changing the shape of the onset cue as well. The stronger miscue results in the platform velocity being reduced quickly to zero and the platform does not consequently utilising the full workspace during slow down.

### B. Model Predictive Control

In MPC a constrained finite-horizon optimisation problem is solved at every time step. Since this is a receding horizon technique, only the control action at the first-step is applied to the system. The numerical solution is found by reformulating the optimisation problem as a quadratic programming problem, which is solved using standard algorithms [26].

The cost function of the motion cueing problem is:

$$J = \sum_{i=0}^{H_P} y^T(t+i)Qy(t+i) + u^T(t+i)Ru(t+i), \quad (34)$$

where  $R$  and  $Q$  are weighting matrices,  $u(t+i)$  is the simulator acceleration (control action) and  $y(t+i)$  is defined as:

$$y(t+i) = x(t+i) - x_{ref}(t) \quad i = 0, \dots, H_P. \quad (35)$$

In (35)  $x_{ref}(t)$  is a constant reference and  $x(t+i)$  are the system states that evolve according to the linear plant dynamics:

$$x(t+i+1) = Ax(t+i) + Bu(t+i) \quad i = 0, \dots, H_P, \quad (36)$$

The problem constraints are:

$$u_{min} \leq u(t+i) \leq u_{max} \quad i = 0, \dots, H_U - 1 \quad (37)$$

$$v_{min} \leq v_{plat}(t+i) \leq v_{max} \quad i = 0, \dots, H_P \quad (38)$$

$$s_{min} \leq s_{plat}(t+i) \leq s_{max} \quad i = 0, \dots, H_P \quad (39)$$

$$u(t+i) = 0 \quad i = H_U, \dots, H_P \quad (40)$$

In equations (34) to (40)  $H_P$  is the prediction horizon (the number of time-steps over which the cost function is minimised) and  $H_U$  is the control horizon (the number of time steps for which the control action is solved). Since  $H_U \leq H_P$  the control is assumed zero from  $H_U$  to  $H_P$ , equation (40). The state vector  $x(t+1)$  contains the vestibular states and the platform actuator lengths, velocities and accelerations. These actuator motions are computed using a linearised platform model, based on small-angle approximations, which is described in [9]. The actuator movements are limited in constraints (37), (38) and (39). The error given in (35) and the platform acceleration  $u(t+i)$  both appear in the cost function (34), and so both are penalised. As one would expect, the strength and smoothness of the onset and miscue in the acceleration signal are affected by the length of control and prediction horizons.

If  $H_U$  is held constant, then increasing  $H_P$  causes an early acceleration washout and a reduced peak in the onset acceleration. This behaviour results from the fact that a longer prediction horizon leads to early anticipation of a velocity constraint violation and the platform acceleration is reduced accordingly.

For a constant  $H_P$ , decreasing  $H_U$  results in a sharper onset acceleration. It follows from (40) that a shorter  $H_U$  results in a reduced time window within which the cost can be minimised. In order to reduce the perception error, the platform is driven at maximum acceleration for the short period allowed.

Iterative tuning allows a desired onset cue to be achieved, but the shape of the miscue may not be acceptable. To achieve an acceptable compromise, one exploits the fact that longer prediction and control horizons ‘smooth out’ the miscue, although this tends to reduce the peak acceleration obtained.

As seen in Figure 20, the MPC cueing signal exhibits an onset cue that is similar to the LQR and NOC responses. The actuator limits are respected, since they are constrained by the quadratic programme. The platform utilises the full workspace. As one can see, there is a strong miscue late in the braking phase that brings the platform out of velocity saturation.

### C. Remarks

The comparative study of these three motion cueing algorithms has identified several issues.

- 1) The actuator velocity constraints limit the magnitude and duration of the onset cue. If this cue is strengthened, it must also be made shorter in order to prevent velocity saturation;
- 2) the miscue that slows the platform as it approaches the workspace limits is a by-product of these constraints. This miscue can nevertheless be reshaped.
- 3) The NOC achieved a smaller miscue than the LQG and MPC methods. This indicates that it is possible to improve the performance of these approaches, perhaps by introducing non-linear features to the algorithms.
- 4) The miscue is smaller if the platform is not needlessly re-centred as in the LQG case. The inclusion of prepositioning in all of the algorithms will improve the miscue performance.
- 5) In the LQG and MPC methods the miscue and onset characteristics cannot be tuned independently.
- 6) The increased flexibility available in the NOC framework allows the acceleration cueing and miscue

management phases to be dealt with separately. During onset cueing the perception error is minimised, while during the miscue phase the platform acceleration is reduced. Improvements in the LQG and MPC performance are in prospect if the separate adjustment of the two cueing phases were made possible.

The next section of this work examines the modification of the LQG strategy to improve the false cue characteristics.

## IX. LQG EXTENSIONS

The linear filtering approach is now extended to include two non-linear post-filtering modules. The first is the anti-backlash filter described in [6]. The performance of this algorithm is compared with an alternative miscue-resaping procedure that is described in section IX-B.

### A. Anti-backlash Filter

The system designed in [6] is illustrated in Figure 21. This additional module applies an adaptive gain to the output of the cueing filter and reduces the magnitude of the acceleration cue during false cueing. The acceleration reference signal,  $a_{ref}^x$ , is filtered by the high-pass LQG-designed cueing filter,  $F_{HP}(s)$ . The adaptive gain  $G$ , given by

$$G = \exp(-k|\alpha - a_{ref}^x|), \quad (41)$$

is applied to the filtered output  $\alpha$ ;  $k$  is a tuning parameter. The output of the adaptive gain  $\beta$ , the unfiltered signal  $\alpha$ , together with the reference acceleration are inputs to a switching function  $S(\alpha, \beta, a_{ref}^x)$ . The switching function is given by:

$$\gamma = (1 - K)\alpha + K\beta, \quad (42)$$

in which  $K$  is a smooth switching function given by:

$$K = 0.5(1 + \tanh(|\alpha - a_{ref}^x| - |\beta - a_{ref}^x|)). \quad (43)$$

The switch toggles between 0 and 1 as the argument of the hyperbolic-tangent function changes from negative to positive. Therefore, the output  $\gamma$  switches from  $\alpha$  to  $\beta$  so as to achieve least error.

The cueing filter is required to slow the platform as well as washout the acceleration demand. The miscue reduction produced by the switch  $S(\alpha, \beta, a_{ref}^x)$  may produce a non-zero platform terminal velocity. To correct for defect, the output  $\gamma$  is passed through a high-pass re-centring filter  $F_{RC}(s)$  that assists in maintaining the platform within its workspace.

### B. Miscue Resaping Module

The anti-backlash approach described in the previous section does not include prepositioning. Additionally, an extra re-centring filter is required, which affects the shape of the onset cues. To address these issues another non-linear module has been designed with the following requirements:

- 1) The additional module must not change the acceleration onset cues;
- 2) The false cues must have lower magnitude, with necessarily longer duration than the standard LQG filter;
- 3) The platform must be prepositioned before a manoeuvre;
- 4) The shape of the onset cues and false cues should be easily, and independently, adjustable.

A system shown in Figure 22 fulfils these requirements. The reference acceleration  $a_{ref}^x$  is filtered by  $F_{HP}(s)$  to produce an output  $\alpha$ . The filter is second-order, high-pass and designed to produce large-magnitude onset cues that do not exceed workspace limits.

The output of the feedback controller  $G(s)$  is:

$$\zeta = K_1 v_{sim}^x + K_2 (p_{ref} - p_{sim}^x). \quad (44)$$

where  $\zeta$  is an acceleration signal that drives the platform velocity to zero and the position to  $p_{ref}$ . The reference position is  $+1$  when the vehicle is accelerating,  $-1$  during braking and the platform is prepositioned for the next manoeuvre. The selection of the feedback gains  $K_1$  and  $K_2$  in (44) influences the shape of the acceleration trajectory. The switching function  $S(\alpha, \beta, a_{ref}^x)$  produces the acceleration cue as before.

In the onset phase of a manoeuvre  $a_{sim}^x = \alpha$  and so the simulator acceleration demand is the high-pass filtered reference signal. After this the onset cue  $a_{sim}^x = \beta$  and the platform is slowed to a halt at the reference position. The switching function detects the beginning of a manoeuvre through the sign change in the vehicle's acceleration. At this time the output is set to  $\alpha$  and after  $\Delta t$  seconds the output switches to  $\beta$ . The delay  $\Delta t$  is chosen to maximise the duration of the miscue phase without interfering with the onset cue.

## X. LQG EXTENSION RESULTS

The two miscue reduction techniques are compared with the standard LQG response in Figures 23 and 24. In this study the acceleration reference is produced by a high-fidelity car model and contains the high-frequency information necessary for a detailed comparison of each filtering method. The adaptive gain in the anti-backlash filter is tuned to reduce the false cues, whilst the re-centring filter was adjusted to ensure compliance with the actuator limits.

In the acceleration responses shown in Figure 23, it is clear that the anti-backlash system response (green) is attenuating the miscue present in the standard LQG response (blue). However, all the high-frequency information in the reference acceleration is being lost. This loss of high-frequency information is unnecessary, because the associated motions are small; it is also inconsistent with the rest of the acceleration signal. Referring to Figure 24, one observes that unlike the standard LQG approach the re-centring filter  $F_{RC}(s)$  has reset the platform velocity within the braking period. It is also clear from this figure that this system does not utilise the full platform workspace, although it is an improvement over the standard LQG filter. As with the standard LQG filter, the anti-backlash approach does not bring the platform to a halt within the braking period.

The high-pass cueing filter in the return-mode controller was tuned to produce a strong onset cue (red), with the feedback controller adjusted to ensure the platform is correctly positioned for the subsequent acceleration manoeuvre. In contrast to the anti-backlash system, the miscue-reshaping module preserves the high-frequency information in the acceleration reference. The miscue associated with the miscue-reshaping module is weaker than the standard LQG response, but worse than the anti-backlash method. This new miscue reshaping module shown in Figure 22 makes full use of the displacement workspace through the inclusion of prepositioning.

## XI. CONCLUSION

Race car simulators present a different set of problems as compared with simulators for passenger cars, with high longitudinal decelerations being a particularly demanding challenge in the race car context. When cueing braking manoeuvres it is especially important to identify the cues the drivers need and to use the workspace to its full capacity.

Existing motion cueing algorithms do not consider explicitly the optimal usage of the workspace. Linear filtering is constrained by linearity and quadratic performance indices. While MPC observes the physical constraints of the system, system linearity is assumed and the system performance is heavily influenced by the control and prediction horizons. By addressing the cueing problem in a numerical optimal control framework, it is possible to overcome these limitations and the workspace usage is easily analysed. Time-varying cost functions can be used to include driver preferences such as strong timely onset cues.

In an initial study, motion cueing in the longitudinal, lateral and yaw directions were considered independently. Actuator velocity limits were found to be the most important influence affecting simulator performance with workspace volume also influential. Thereafter, the full 3-DOF motion system was analysed and the coupling between the degrees of freedom discussed. In particular, yaw rotation causes a substantial reduction in the translational workspace. As a result, the performance of the simulator in the yaw freedom must be compromised in order to accommodate translational performance.

The numerical optimal control solution for a longitudinal braking manoeuvre was compared with the MPC and LQG techniques. While the onset cue performances can be tuned to be similar, the miscues produced by both the MPC and LQG systems are worse than those associated with NOC. The inability to adjust miscues and onset cues separately was identified as a critical limitation associated with these methods.

In order to improve the LQG false cue performance two non-linear modules were tested. The anti-backlash filter was found to reduce the miscue, but it has a detrimental effect on high-frequency performance. A new miscue reshaping module showed improved miscue and workspace usage performance.

Nonlinear numerical optimal control has been established as an effective tool for producing baseline performance standards for other more conventional simulator cueing techniques. Although NOC is not a feedback strategy, it is a useful way of analysing the performance limits of other driver-in-the-loop strategies.

## REFERENCES

- [1] J. Dichgans and T. Brandt, "Visual-vestibular interaction: Effects on self-motion perception and postural control," in *Handbook of Sensory Physiology*. Springer Berlin Heidelberg, 1978, vol. 8, pp. 755–804.
- [2] S. Schmidt and B. Conrad, "Motion drive signals for piloted flight simulators," Washington: National Aeronautics and Space Administration, Tech. Rep. NASA CR-1601, 1970.
- [3] A. Capustiac, D. Banabic, D. Schramm, and U. Ossendoth, "Motion cueing: From design until implementation," In Proceedings of Romanian Academy, Series A, Vol.12 No.3, Romanian Academy, 2011.
- [4] W. Krissada, Chundang & Wannasuphoprasit, "Washout filter for a 3 DOF simulator," in *The 5th International Conference on Automotive Engineering*, 2009.
- [5] F. Colombet, M. Dagdalen, G. Reymond, C. Pere, F. Merienne, and A. Kemeny, "Motion cueing: What is the impact on the driver's behaviour?" In Proceedings of the Driving Simulation Conference Europe, Monaco, France, February 2008.
- [6] G. Reymond and A. Kemeny, "Motion cueing in the Renault driving simulator," *Vehicle System Dynamics*, vol. 34, No.4, pp. 249–259, 2000.
- [7] R. Telban, W. Wu, and F. Cardullo, "Motion cueing algorithm development: Initial investigations and redesign of the algorithms," State University of New York, Tech. Rep. CR-2000-209863, 2000.
- [8] M. Nahon and L. Reid, "Simulator motion-drive algorithms: A design's perspective," *Journal of Guidance, Control and Dynamics*, vol. 13, No.2, pp. 356–362, 1990.
- [9] N. Garrett and M. Best, "Driving simulator motion cueing algorithms ? a survey of the state of the art," In Proceedings of the 10th International Symposium on Advanced Vehicle Control (AVEC), Loughborough, UK, August 2010.
- [10] L. Nehaoua, H. Mohellebi, A. Amouri, H. Arioui, S. Espi, and A. Kheddar, "Design and control of a small-clearance driving simulator," *IEEE Transactions on Vehicular Technology*, vol. 56, No.2, pp. 736–746, 2008.
- [11] D. Ariel and R. Sivan, "False cue reduction in moving flight simulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-14, No.4, pp. 665–671, 1984.
- [12] R. Telban, F. Cardullo, and J. Houck, "Developments in human centered cueing algorithms for control of flight simulator motion systems," *AIAA Modeling and Simulation Technologies Conference*, 1999, Portland, OR. Paper Number AIAA-99-4328.
- [13] R. Sivan, J. Ish-Shalom, and J. Huang, "An optimal control approach to the design of moving flight simulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 12, iss. 6, pp. 818–827, 1982.
- [14] J. Ish-Shalom, "Design of optimal motion for flight simulators," Ph.D. dissertation, Massachusetts Institute of Technology, 1982.
- [15] C. Weiß, "Control of a dynamic driving simulator: Time-variant motion cueing algorithm and prepositioning," Ph.D. dissertation, Deutsches Zentrum für Luft und Raumfahrt, 2006.

- [16] T. Chapron and J.-P. Colinot, “The new PSA Peugeot-Citroën advanced driving simulator overall design and motion cue algorithm,” in *Driving Simulation Conference 2007, North America, Iowa City*, 2007.
- [17] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maizi, “Model-based predictive motion cueing strategy for vehicle driving simulators,” *Control Engineering Practice*, vol. 17, pp. 995–1003, 2009.
- [18] G. Perantoni and D. J. Limebeer, “Optimal control for a formula one car with variable parameters,” *Vehicle System Dynamics*, vol. 52, iss. 5, pp. 1–27, 2014.
- [19] D. J. N. Limebeer, G. Perantoni, and A. V. Rao, “Optimal control of formula one car energy recovery systems,” *International Journal of Control*, 2014, published online: 23 Apr 2014. DOI 10.1080/00207179.2014.900705.
- [20] B. Dasgupta and T. Mruthyunjaya, “A Newton-Euler formulation for the inverse dynamics of the Stewart platform manipulator,” *Mechanical Machine Theory*, vol. 33, no. 8, pp. 1135–1152, 1998.
- [21] J. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, ser. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2001.
- [22] M. A. Patterson and A. V. Rao, *GPOPS II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming*, University of Florida, ACM Trans. Math. Soft. 39, 3, Article 1, July 2013.
- [23] D. Garg, M. A. Patterson, W. W. Hager, A. V. Rao, A. V. Benson, and G. T. Huntington, “A unified framework for the numerical solution of optimal control problems using pseudospectral methods,” *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.
- [24] J. Meiry, “The vestibular system and human dynamic space orientation,” Ph.D. dissertation, Massachusetts Institute of Technology, 1965.
- [25] M. Green and D. J. N. Limebeer, *Linear Robust Control*. Mineola, New York: Dover, 2012.
- [26] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.





Ingrid Salisbury received a B.Sc. degree in electronic engineering from the University of Kwazulu-Natal, Durban, South Africa in 2011. She is currently working towards the degree of Doctor of Philosophy in engineering science at the University of Oxford, Oxford, United Kingdom.

Her current research interests include race-car driving simulators, motion cueing algorithms, optimal control and vehicle dynamics and stability.



**David J N Limebeer** received the B.Sc.(Eng) degree from the University of the Witwatersrand in 1974, MSc(Eng) and PhD degrees from the University of Natal in 1977 and 1980, respectively, and the DSc (Eng) from the University of London in 1992. After completing his PhD degree he was a post-doc researcher at the University of Cambridge between 1980 and 1984. He then joined the Electrical and Electronic Engineering Department at Imperial College as a lecturer. He was promoted to Reader in 1989, Professor in 1993, Head of the Control Group in 1996, and Head of Department 1999-2009. In 2009 he moved to Oxford as Professor of Control Engineering and Professorial Fellow at New College Oxford. His research interests include a range of applied and theoretical problems in control systems and engineering dynamics including: robust control, optimal control, process control and the control of aeroelastic phenomena in large structures. He is also interested in a variety of problems in multibody mechanics and the dynamics of two- and four-wheeled road vehicles. He is a Fellow of the IEEE (1992), a Fellow of the IET (1994), a Fellow of the Royal Academy of Engineering (1997) and a Fellow of the City and Guilds of London Institute (2002). His consultancy activities have been in a variety of vehicle-related matters, product liability litigation, and patent disputes in optical recording systems, drilling equipment and high-speed packing machines.

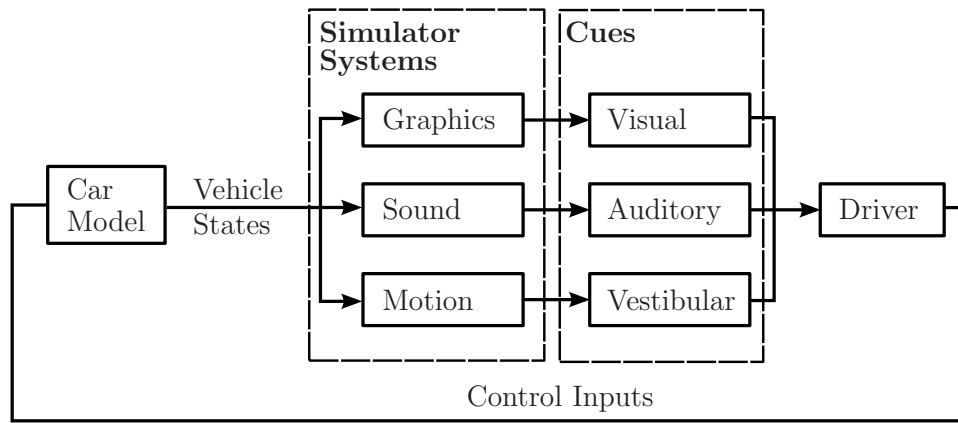


Fig. 1. Components of a generic vehicle simulator and their relationship to the driver.

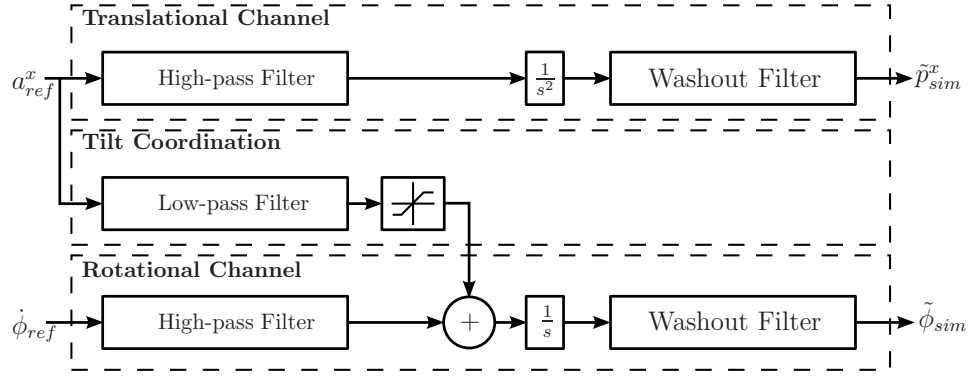


Fig. 2. Structure of the classical cueing system with tilt-coordination.

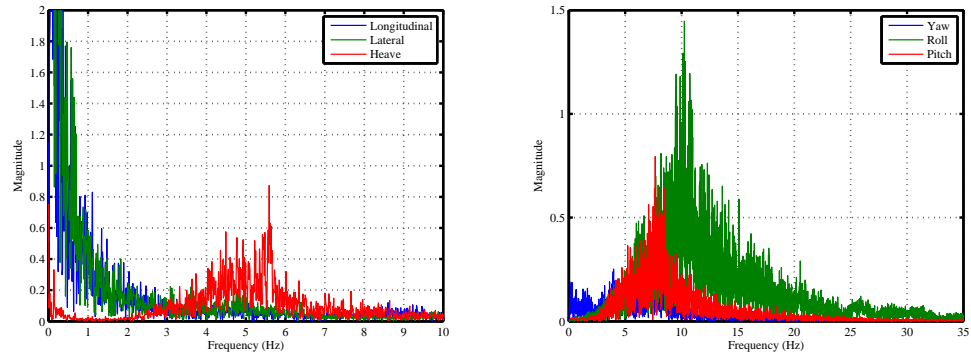


Fig. 3. The left- and right-hand figure contain the frequency responses of the translational and rotational acceleration signals respectively.

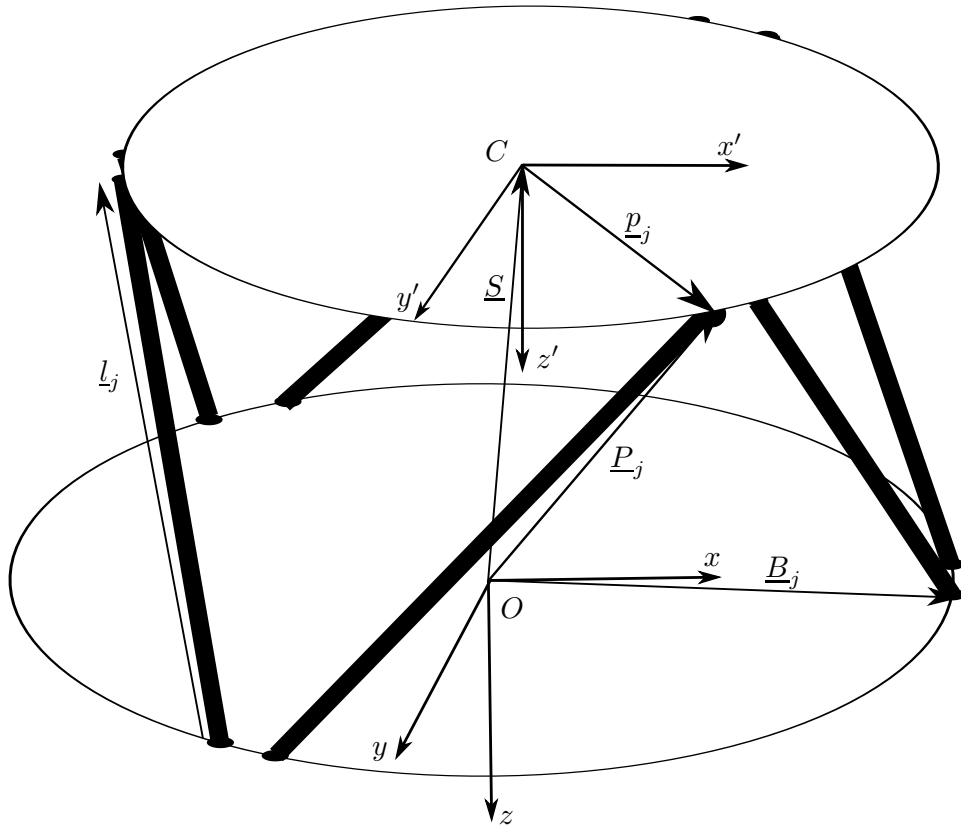


Fig. 4. Hexapod motion platform with the body-fixed and inertial reference frames, and leg, base-joint and platform-joint vectors.

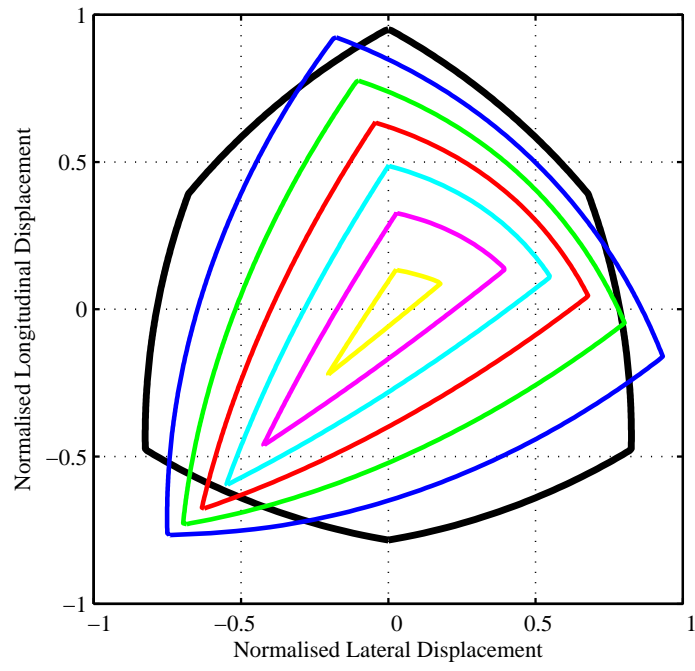


Fig. 5. Body-fixed lateral and longitudinal hexapod workspace boundaries for increasing yaw angles from  $0^\circ$  (black) to  $30^\circ$  (yellow) in steps of  $5^\circ$ .

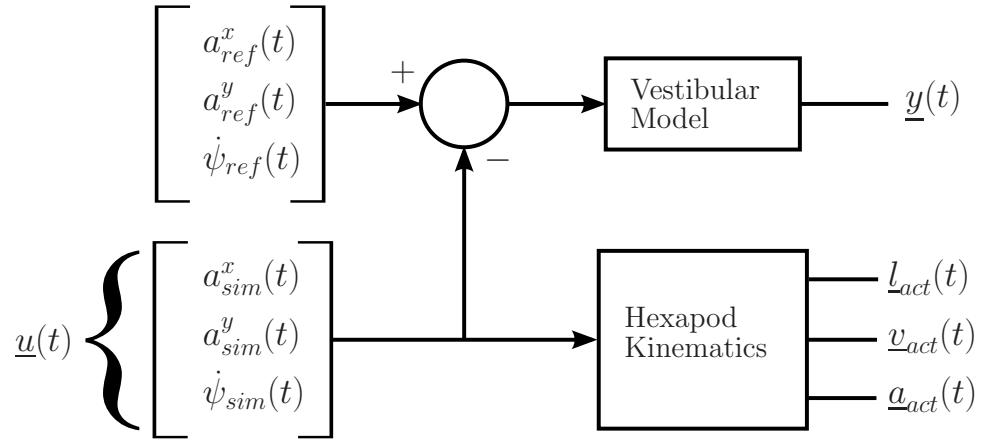


Fig. 6. Formulation of numerical optimal control problem.



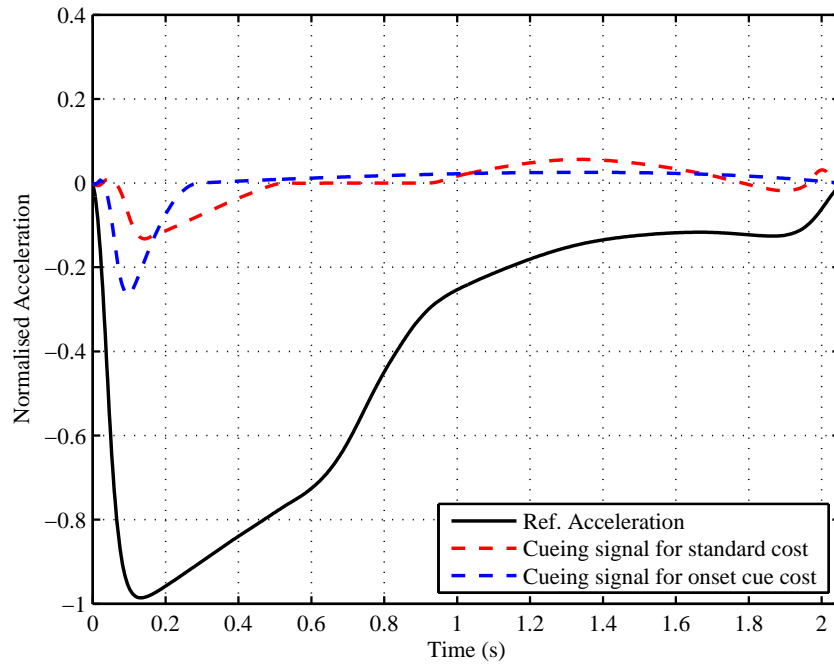


Fig. 7. Simulator longitudinal acceleration demand over a braking manoeuvre as determined by the open-loop numerical optimal control simulation for a standard cost function (19), and a cost function that emphasises onset cues (26).

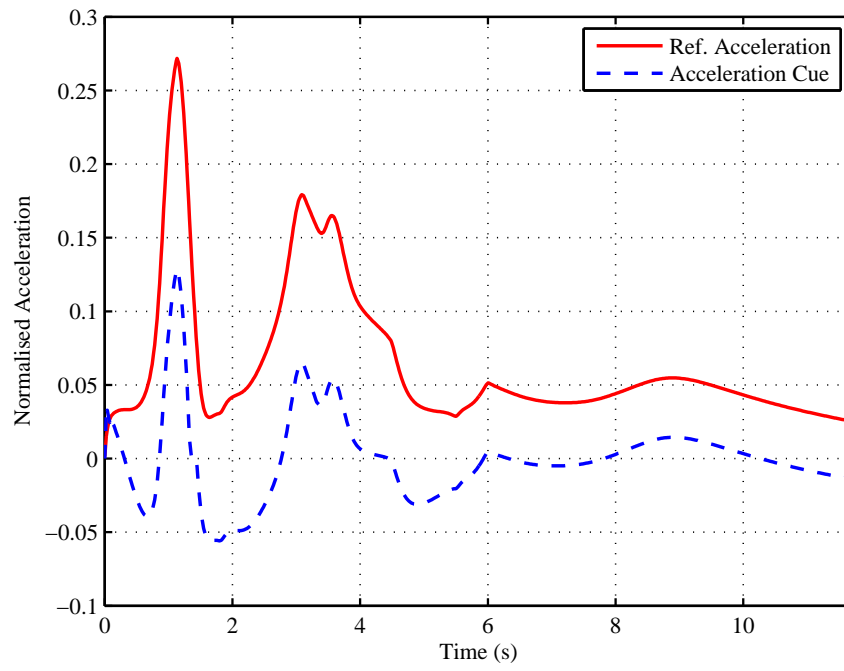


Fig. 8. Results of open-loop numerical optimal control simulation for a (longitudinal) accelerating manoeuvre.

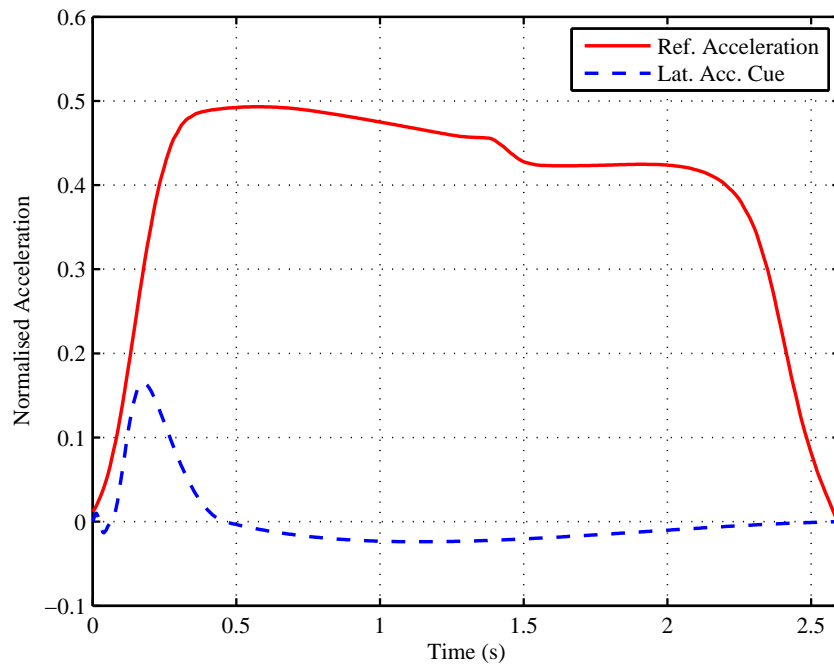


Fig. 9. Results of open-loop numerical optimal control simulation for a lateral acceleration in a high-speed corner.

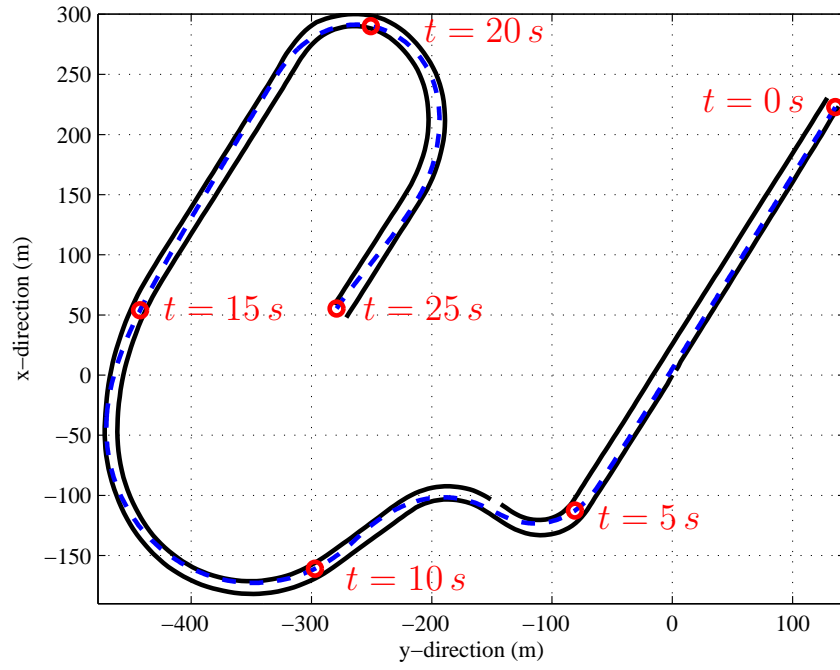


Fig. 10. Track and racing line for the first four corners of the Barcelona circuit used in numerical optimal control simulations [18, 19]. The red markers are spaced 5 seconds apart, with the first being at 0 seconds and the last at 25; the car travels in a clockwise direction.

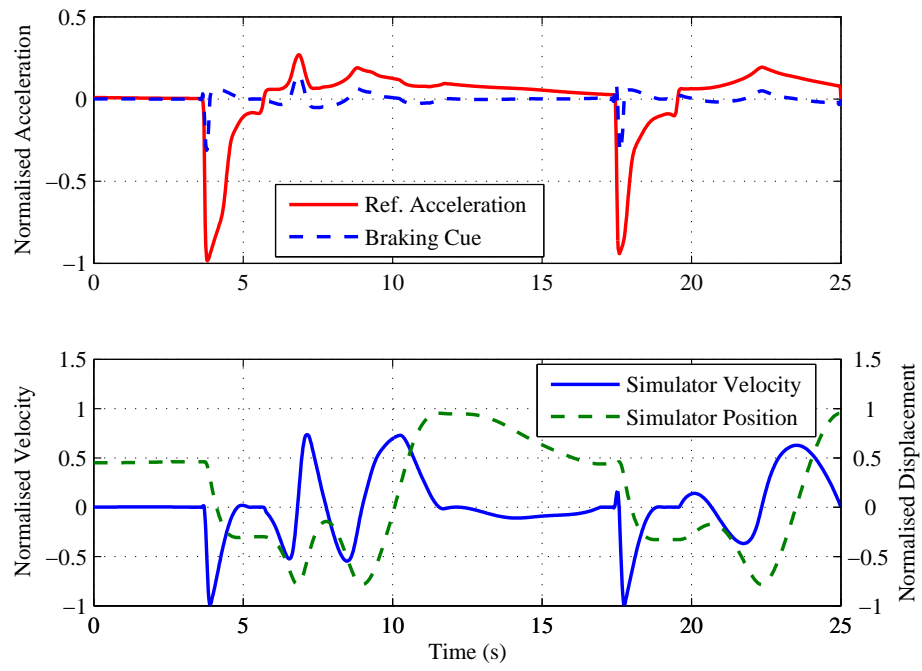


Fig. 11. The top figure shows the longitudinal acceleration cueing signal determined by the numerical optimal control computation for the given longitudinal acceleration reference. The lower figure shows the corresponding simulator velocity and position demand.

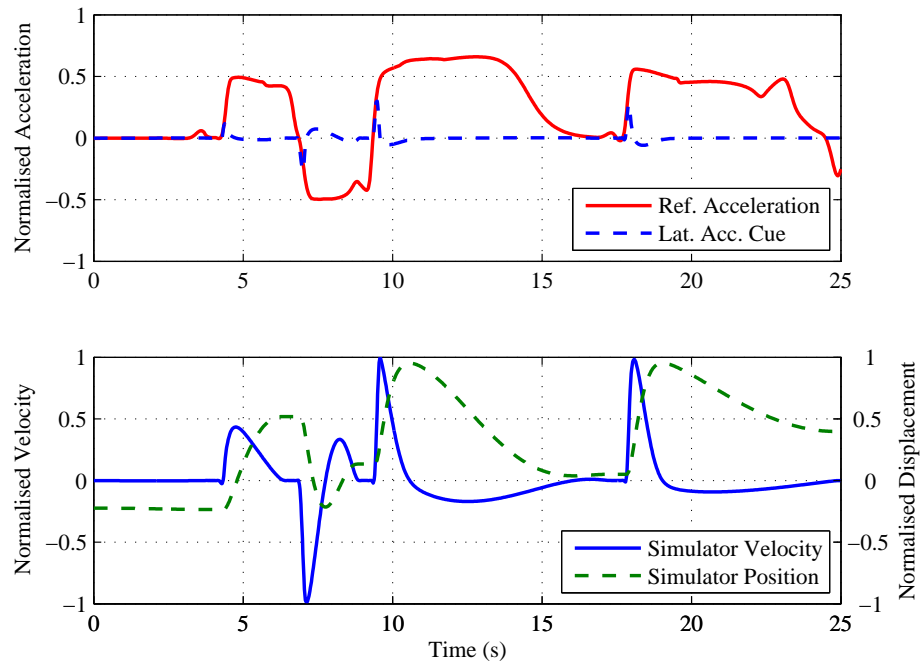


Fig. 12. The top figure displays the lateral acceleration cueing signal determined by the numerical optimal control computation for the given lateral acceleration reference. The lower figure shows the corresponding simulator velocity and position demand.

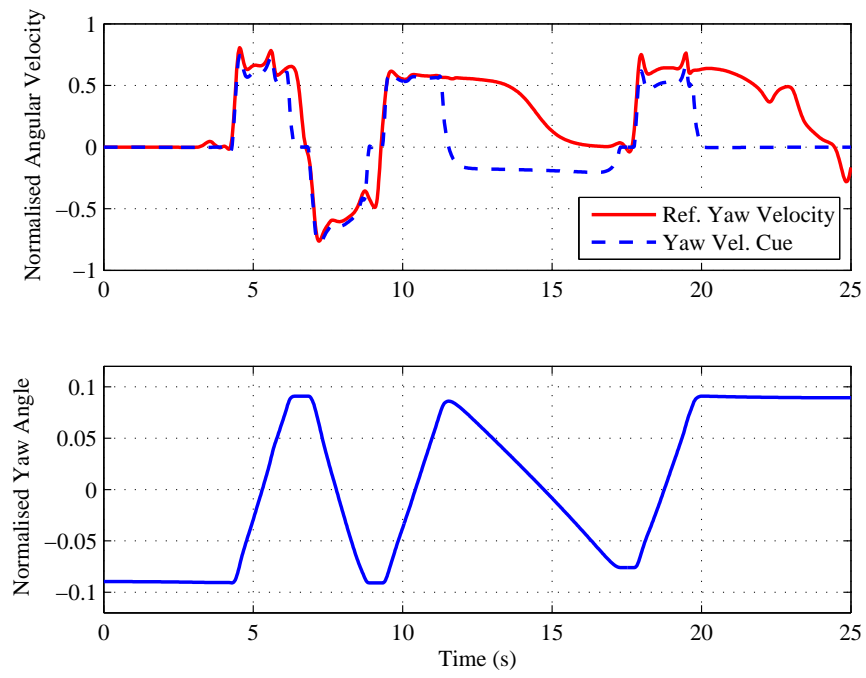


Fig. 13. The top figure shows the yaw velocity reference and cueing signal computed by the numerical optimal control solver. The lower figure shows the associated platform rotational position demand.

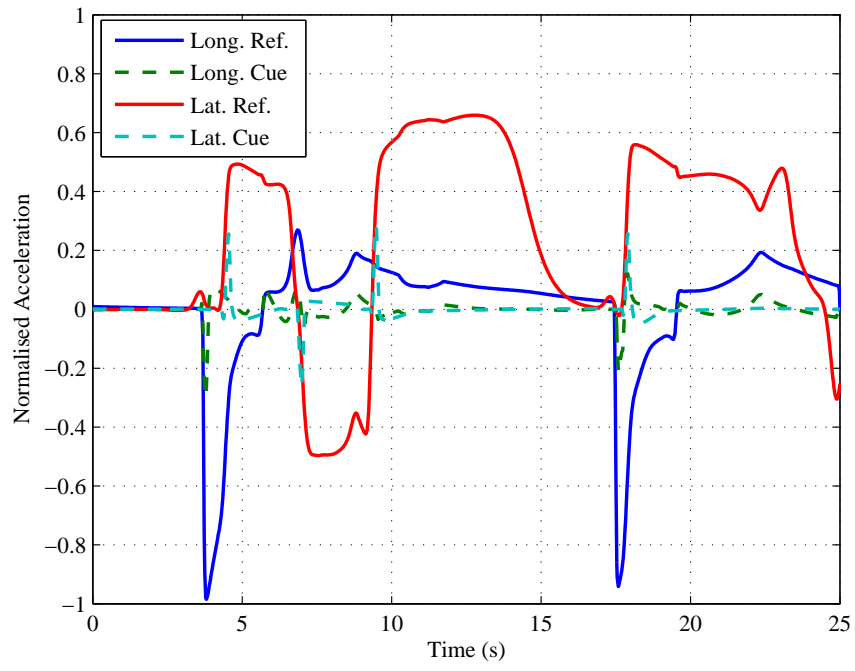


Fig. 14. The longitudinal and lateral platform acceleration demands computed using the numerical optimal control solver for given reference accelerations.



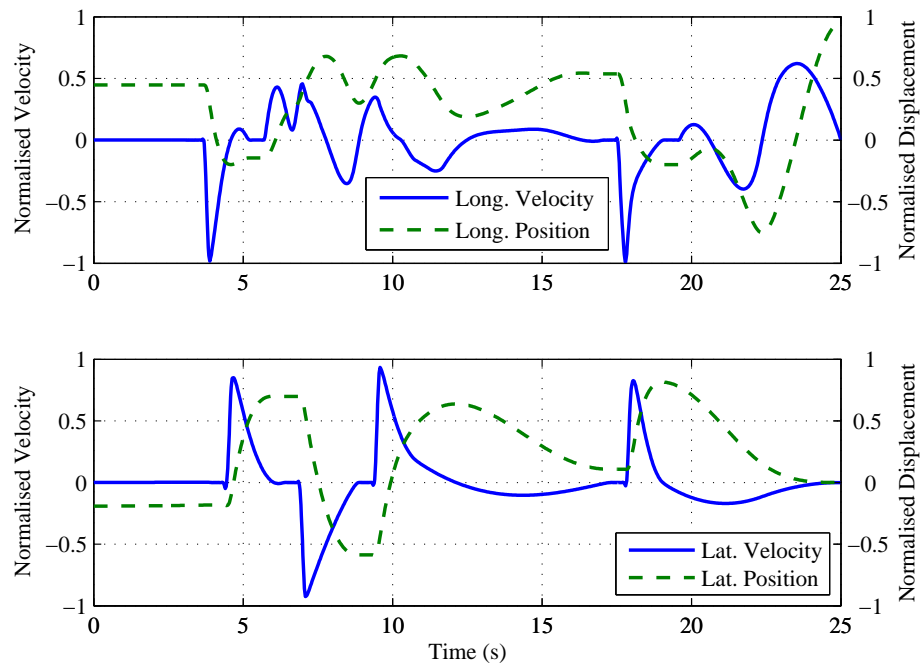


Fig. 15. The simulator velocity and position demands for the longitudinal (top figure) and lateral (bottom figure) directions. These are associated with the acceleration demands given in Figure 14.

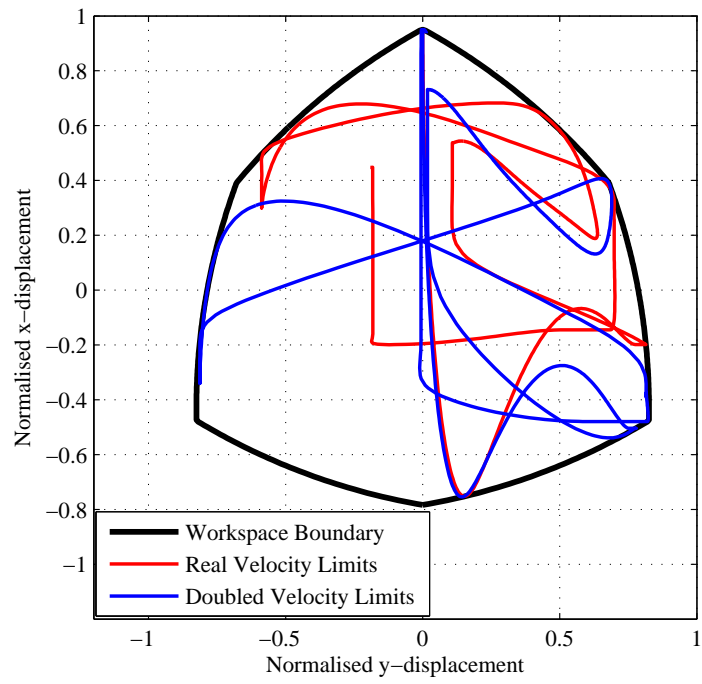


Fig. 16. The platform boundary and the use of the workspace over the four corners for the actual actuator velocity limit (red) and a doubled actuator velocity limit (blue).

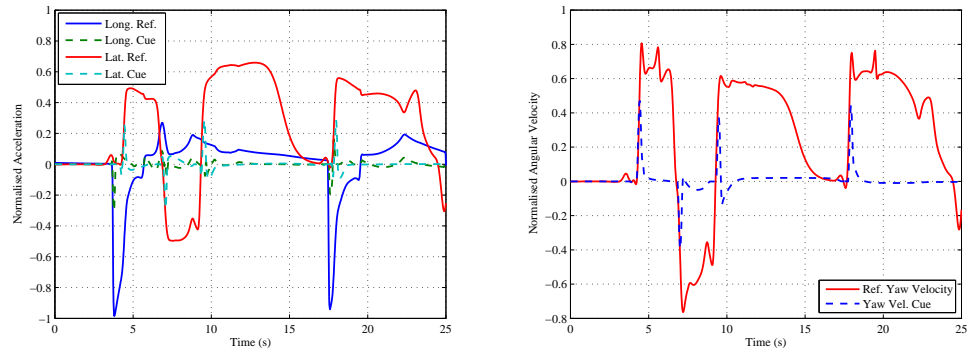


Fig. 17. Longitudinal (green-dashed) and lateral (light-blue dashed) acceleration, and yaw velocity platform demands computed by numerical optimal control. The left-hand figure shows the longitudinal and lateral acceleration responses, while the right-hand figure shows the yaw velocity response.

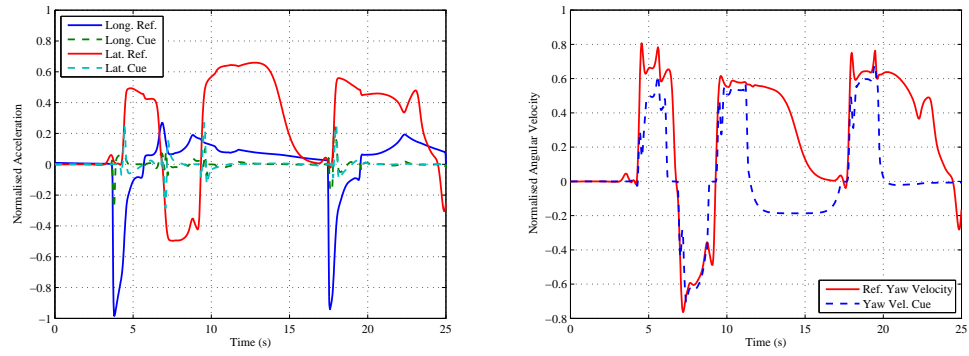


Fig. 18. Longitudinal (green-dashed) and lateral (light-blue dashed) acceleration, and yaw velocity platform demands computed by numerical optimal control. The left-hand figure shows the longitudinal and lateral acceleration responses, while the right-hand figure shows the yaw velocity response.

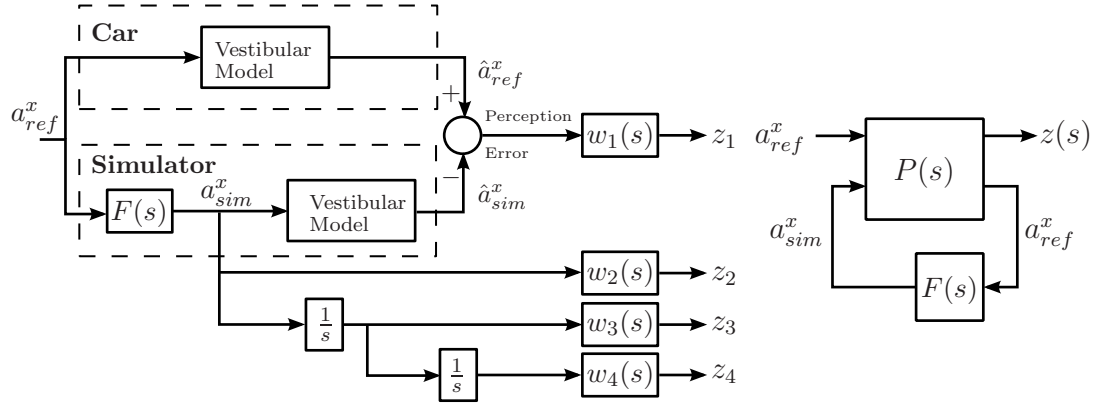


Fig. 19. LQG filter design framework. The left-hand figure shows the LQG cueing system and the right-hand figure shows the embedding in a generalised regulator format [25].

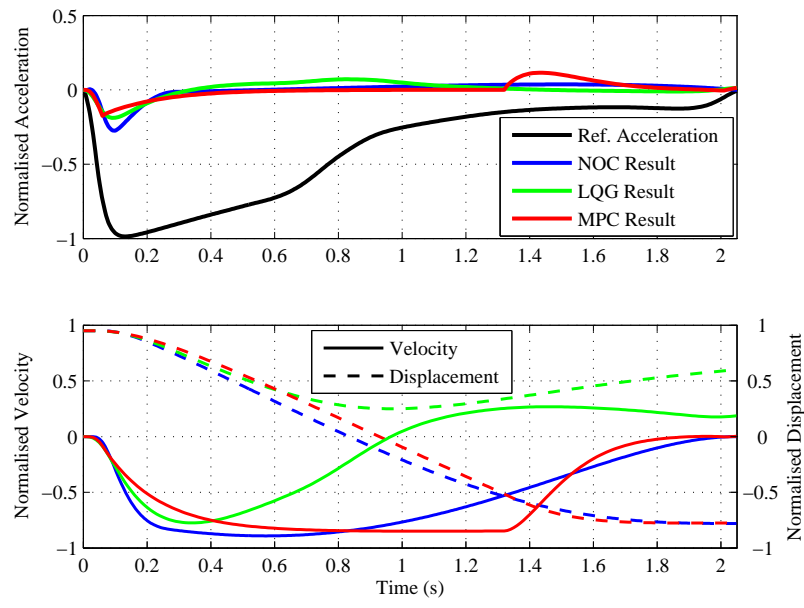


Fig. 20. Comparison of numerical optimal control, linear quadratic gaussian and model predictive control cueing strategies over a typical longitudinal braking manoeuvre.

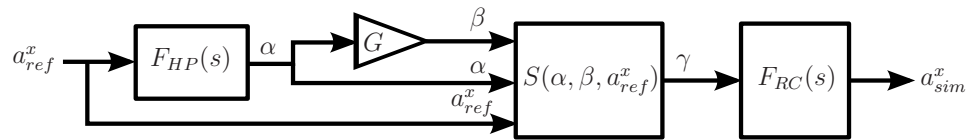


Fig. 21. Optimal filter cueing structure with an additional anti-backlash module.

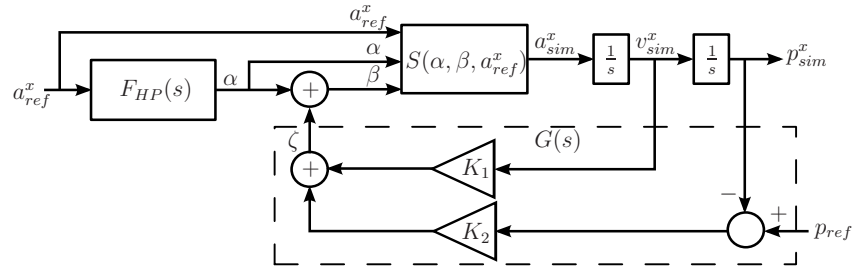


Fig. 22. Structure of the cueing filter with a miscue reshaping module.



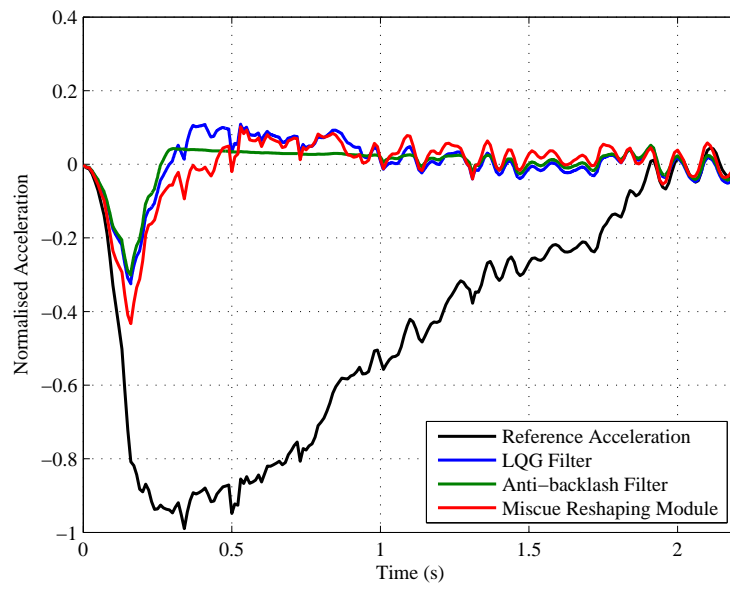


Fig. 23. Longitudinal acceleration demand for standard LQG, the anti-backlash and the miscue reshaping technique.

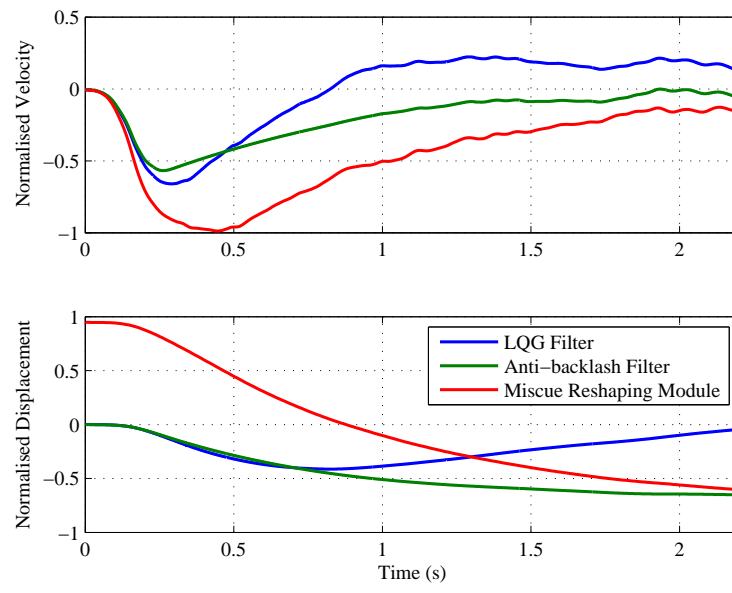


Fig. 24. Position and velocity demands for the standard LQG, anti-backlash and the miscue reshaping technique.