

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/>

Driving Simulator Motion Cueing Algorithms – A Survey of the State of the Art

Nikhil J I Garrett and Matthew C Best
Loughborough University, UK

Dept. Aeronautical and Automotive Engineering, Stewart Miller Building, Loughborough University,
Loughborough, Leicestershire, LE11 3TU, UK
Phone: +44 1509 227264
Fax: +44 1509 227275
E-mail: N.J.I.Garrett@lboro.ac.uk

This paper reviews the state-of-the-art motion cueing algorithms for motion-based driving simulators. The motion cueing problem is presented, together with the main published algorithms – classical washout filtering, adaptive filtering, linear optimal control, and model predictive control (MPC). Implementation details for each of the algorithms are given and their response to various manoeuvres plotted. The algorithms all have a high-pass response apart from the MPC algorithm, which reproduces vehicle motion for as long as possible before returning to centre. The cost function-based algorithms require more parameters to be tuned, but the parameters have more relevance to the simulator operator and are thus easier to tune. Finally, proposals for an algorithm evaluation study with human test drivers are given, the results of which will be used in future work to develop a new driving simulator cueing algorithm.

Topics / Driving Simulator, State of the Art & Survey

1. INTRODUCTION

Technological advances in the past few decades mean that driving simulators can now achieve impressive levels of realism; graphics systems provide photo-realistic images, real-time vehicle models replicate vehicle behaviour to a high degree of accuracy, and high-fidelity steering torque motors can reproduce exactly the steering feel of the real vehicle. Many mid- to high-end simulators also include some form of motion platform, the most common of which is the 6-Degree of Freedom (DOF) Stewart platform. However, the problem of how best to generate motion cues from the simulated vehicle motion remains the subject of some discussion, with many different algorithms described in the literature.

The research being carried out at Loughborough University aims to find an improved motion cueing algorithm for driving simulation, in particular one which provides the best possible feedback about the vehicle state to the driver and is easy for a non-expert to tune. In this paper the initial review of the state-of-the-art is presented, along with the plan for the next stage of the research – a simulator study with human test subjects that will assess the performance of the various published algorithms.

2. MOTION CUEING ALGORITHMS

The motion platforms used in driving simulators,

for example the 6-DOF Stewart platform of the Loughborough simulator (Fig. 1), tend to have a very limited motion workspace (in the case of the Loughborough simulator, translational and rotational motion limits are of the order $\pm 0.5\text{m}$ and $\pm 20^\circ$ respectively). In general, the range of motion of a road vehicle in a normal manoeuvre far exceeds the available motion workspace. Thus some transformation is needed which calculates a realizable set of platform motions from the simulated vehicle motion. This transformation is known as the motion cueing algorithm or the washout algorithm.

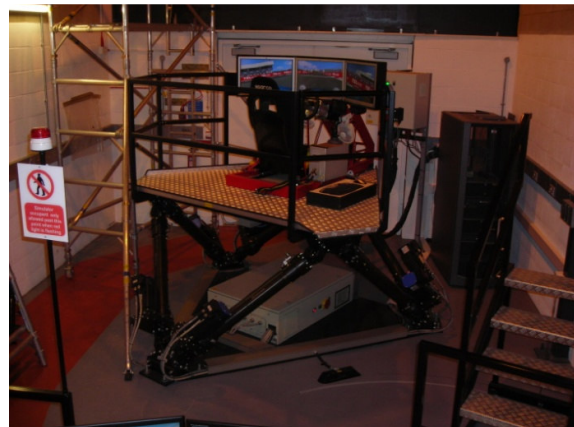


Fig. 1 – Loughborough AAE Dept. Driving Simulator

Historically, motion-based flight simulation has been much more popular than driving simulation due to the advantages that even the early expensive and low-quality flight simulators had over real flight testing. Thus much of the development of motion cueing algorithms has been based around flight simulation rather than driving.

The four most popular published algorithm architectures are described in this section; they are presented in the order in which they were first published. In all of the algorithms, it is the linear accelerations and angular velocities that are filtered.

2.1 Classical washout algorithm

The first motion cueing algorithm described in the literature uses linear high-pass filters, and has come to be known as the Classical algorithm. The high-pass filters remove low-frequency motion content and thus allow reproduction of the higher frequency onset-type motion whilst ensuring that the commanded platform motion does not exceed the platform workspace (provided of course that the filters are appropriately tuned – the filter cut-off frequencies, damping factors and gains are generally tuned for the anticipated worst-case vehicle acceleration). The work of Schmidt and Conrad [1] is one of the earliest published works on the classical algorithm; the later work by Reid and Nahon [2], [3], [4] studies the algorithm in some depth.

Most implementations of the classical algorithm (and, as discussed later, of the other algorithms) use tilt coordination, whereby roll and pitch rotations are used to simulate, respectively, sustained lateral and longitudinal accelerations. Many authors (e.g. Reid and Nahon [2]) argue that the rate of tilt must be below the perception threshold for that particular axis, although Berger et al's results [5] imply that the tilt can be above threshold without significantly affecting the driver performance, provided the visual acceleration correlates with the effective body acceleration.

Figure 2 shows the filter topology for the roll/lateral and pitch/longitudinal axis pairs with tilt coordination.

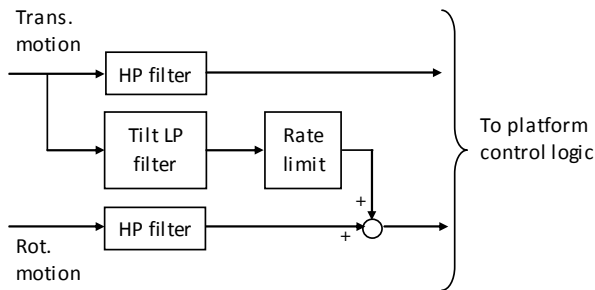


Fig. 2 – Filter topology – tilt-coordination axis pair

The remaining axes (yaw and vertical) are simply high-pass filtered, as there is no coordination between them. Schmidt and Conrad [1] and Reid and Nahon [2] both use 2nd-order filters for translational axes, transfer function as in equation 1:

$$\frac{a_s}{a} = \frac{Ks^2}{s^2 + 2\zeta\omega s + \omega^2} \quad (1)$$

with gain K , cut-off frequency ω and damping factor ζ . Note that the subscript $_s$ denotes a simulator variable (i.e. a_s is the acceleration of the simulator motion platform). The rotational filters are 1st-order, with transfer function of the form of equation 2.

$$\frac{\dot{\theta}_s}{\dot{\theta}} = \frac{Ks}{s + \omega} \quad (2)$$

The low-pass tilt coordination filter is 2nd-order, with the cut-off frequency generally chosen such that the low frequency motion that is attenuated by the high-pass filter gets passed to the tilt coordination channel, i.e. so that all of the translational motion is reproduced either through platform translation or by platform tilt.

It is worth mentioning tuning here – the parameters are generally tuned through a trial-and-error process. Although there are only a few parameters to be tuned for each axis (cut-off frequency, damping and gain), the fact that they bear little relevance to simulator motion means that it is difficult for a non-experienced user to intuitively decide which parameter to vary and in which direction in order to achieve the desired effect.

2.2 Adaptive washout algorithm

An evolution of the classical algorithm employs adaptive filters. The ‘standard’ adaptive scheme, as described by Nahon et al [6] among others, has adaptive filter gains that are varied to minimize a cost function that penalizes motion error (the difference between platform motion and the simulated vehicle motion), motion magnitude and the change in the adaptive parameters from their initial values.

As with the classical algorithm, the pitch/longitudinal and roll/lateral axis pairs are treated together. The tilt low-pass filter gain remains fixed, and the high-pass filter gains are adapted such that the following cost function is minimized:

$$J_y = \frac{1}{2} [W_1(a - a_s)^2 + W_2(\dot{\theta} - \dot{\theta}_s)^2 + W_3\dot{y}_s^2 + W_4y_s^2 + W_5\dot{\theta}_s^2 + W_6\theta_s^2 + \sum_{i=1}^6 W_{i+6}\Delta P_i^2] \quad (3)$$

where the cost weights W_i are used to tune the filters, and P_i are the adaptive filter gains. The yaw and vertical cost functions are similar, but have only the rotational or translational terms respectively.

The steepest descent method is used to move towards the cost function minimum, the adaptive gains adjusted according to equation 4, with the step size K_i available as an additional tuning parameter.

$$\dot{P}_i = -K_i \frac{\partial J}{\partial P_i}, \quad i = 1, \dots, 6 \quad (4)$$

The filter parameters (cut-off frequencies, damping, initial gain values, cost weights and adaptive step size) are again tuned based on the worst-case acceleration. Although the difficulty of how to choose the cut-off frequencies, damping and initial gain values remains the same as the classical algorithm, the cost weights are

much more intuitive for a non-expert to tune; the motion is considered as a trade-off between faithful reproduction of vehicle motion and limiting the platform excursion.

2.3 Optimal control-based algorithm

The optimal control-based algorithm treats motion cueing as a tracking problem – the accelerations perceived in the simulator should track the accelerations that would be perceived in the real vehicle as closely as possible, i.e. minimising $e(s)$ (Figure 3) within the constraints of the motion platform. Such an approach requires some understanding of the relationship between actual body motion and the motion perceived by the brain. Zacharias [7] and Young [8] discuss the various models of the vestibular organs that have been proposed over the years. Different implementations of the optimal algorithm use a variety of different models for the vestibular response; a comprehensive review is not included here, suffice it to say that the different models have similar responses, with the major difference being the order of the models and the exact values of the model parameters.

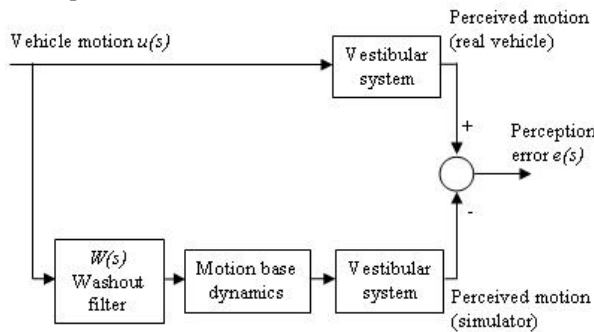


Fig.3 – Motion tracking problem

The optimal algorithm uses a Linear Quadratic Regulator (LQR) tracking formulation to produce the washout filter. This method was developed by Sivan et al [9] and an implementation described by Reid and Nahon [2]. The washout filter that results from the LQR design process is the one that minimizes the cost function of equation 5.

$$J = E\{e^T Q e + x_d^T R_d x_d + u_s^T R u_s\} \quad (5)$$

where e is the perception error, x_d a vector of motion platform states (generally linear displacement and velocity and angular displacement for a tilt-coordinated pair), u_s the platform motion command, and Q , R_d and R are cost weight matrices of appropriate dimension. For the tilt-coordinated axis pairs, the vestibular models include the contribution of head tilt to linear motion perception; the coupling of the axes is thus taken into account during the washout filter design. The resultant washout filter has two inputs and two outputs. The single-axis cases are obviously single-input-single-output; some implementations of the optimal algorithm consider all axes individually, with tilt coordination being performed by the simple low-pass filters of the

classical and adaptive algorithms.

The tuning of the optimal algorithm is done by adjusting the cost function weights. This makes the tuning process even easier for the non-expert; filter parameters are removed completely, and the operator can tune the algorithm purely as a trade-off between motion fidelity and limiting platform excursion.

2.4 Model predictive control algorithm

Recent work by Dagdelen et al [10] proposes an algorithm based on Model Predictive Control (MPC). The algorithm minimizes the perception error ($e(s)$ in Figure 3) whilst remaining within the platform limits. This algorithm has the advantage that it takes the platform limits into account explicitly, thus eliminating the need to tune the algorithm for the worst-case motion.

At each time step, a control sequence is calculated over a horizon N such that the square perception error e^2 is minimized and the platform remains within the workspace limits. The other constraint is that, after two prediction time steps, the platform washes out towards the platform centre below the motion perception threshold over the remainder of the prediction horizon. The first value of the control sequence is used at that time step, and then the process is repeated at each subsequent time step. The effect of this optimization formulation is that the platform motion matches the vehicle motion for as long as possible, then returns to centre when it can no longer do so within the workspace limit.

In order to perform such a computationally expensive process in real time, Dagdelen employs a method where the reachable state set is precalculated such that the problem becomes a single-step optimization. In terms of tuning this is the simplest of the four algorithms discussed here; only the prediction horizon needs to be determined, and this simply needs to be long enough to produce the desired performance i.e. it generally does not need to be tuned for different motion scenarios.

3. ALGORITHM RESPONSE

The motion cueing algorithms described in the previous section have been implemented in MATLAB/Simulink. As a first step in the investigation, the response of the algorithms to various inputs was simulated; the results of a few of these experiments are presented in this section. The results presented here consider a single translational axis only. Note that the cut-off frequency and damping values are the same for the classical and adaptive filters used here, in order to isolate the effect of the adaptive gains.

3.1 Step response

Figure 4 shows the response of the four algorithms to a unit step input. Note that the classical filter produces a false cue between 2s and 6s, i.e. the acceleration of the platform and of the vehicle are in opposite directions. The adaptive algorithm acts to

reduce this false cue by reducing the adaptive gain. The optimal LQR algorithm also exhibits a high-pass type response, the algorithm having been tuned to give a lower peak amplitude but a longer time spent simulating the motion. The MPC response is the most extreme, with the step input being followed exactly for a short period before washing out in the opposite direction. Note that there is a sloped transition between reproduction and washout phases; this is as suggested by Dagdelen et al as a result of their initial tests [10].

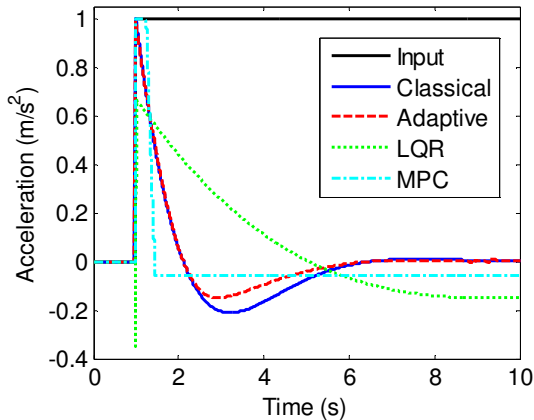


Fig. 4 – Motion cueing algorithm step responses

3.2 Double raised-cosine pulse

Figure 5 shows the response to a double raised-cosine (RC)-sided pulse. The classical and adaptive algorithms show an apparent phase lead, the negative peak in their responses occurring before the negative peak in the input. This is because the filter output does not have a chance to settle before the next motion ‘event’ (e.g. the transition to the negative pulse) occurs. This effect could be tuned out by adjusting the filter cut-off frequency. Note that the adaptive gain reduces during the negative peak in order to reduce the platform excursion.

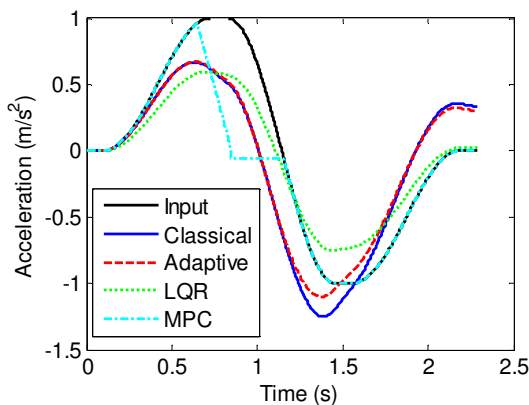


Fig. 5 – Double RC pulse response

The LQR response seems to follow the input quite well, although it must be pointed out that this is more an indicator that the algorithm is tuned well for this

particular manoeuvre than it is an indicator that the LQR algorithm is better than the classical and adaptive algorithms.

The MPC algorithm doesn’t manage to reproduce the first pulse in its entirety, but the fact that the platform is away from centre means that the second pulse can be fully reproduced. This is a strength of this algorithm; it makes full use of the platform workspace, and can therefore take advantage of situations like this.

3.3 Vehicle acceleration data

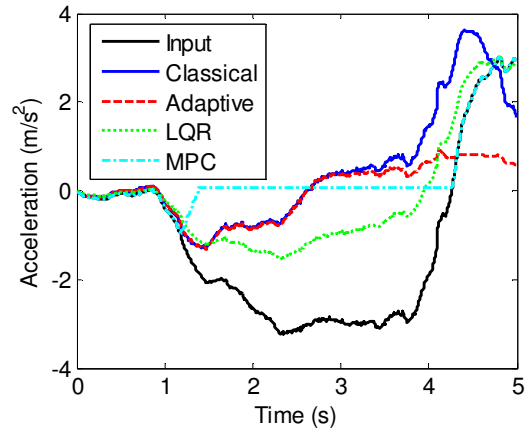


Fig. 6 – Track data response

The results of Figure 6 are from lateral acceleration data from a simulator driving run. Of note here are the false cue in both the classical and adaptive responses at around 2.7s onwards and the strong adaptation of the adaptive gain during the false cue. The slower decay of the LQR response gives less of a false cue, but again this is more down to the tuning set than anything else. This example illustrates a disadvantage of the MPC method; once the algorithm starts to return to centre, none of the high-frequency motion content is reproduced. It is suggested that a period of complete motion reproduction followed by no motion at all (apart from the below-perception-threshold washout) would feel odd to the driver, a theory that will be tested in the simulator study proposed in the next section.

3.4 Discussion

It is inappropriate to draw too many firm conclusions from analysis of simulation results like those above (indeed, over-analysis of simulated results in other work is something the author has been critical of in the past). However some general remarks about algorithm behaviour can be made. The classical algorithm shows reasonable performance but is prone to strong false cues; the adaptive algorithm reduces the magnitude of these false cues and is therefore likely to feel better to the driver. The LQR algorithm also exhibits a high-pass response, so the potential for false cues remains. The MPC algorithm response is significantly different to the other three, but is not necessarily better – the author proposes that ceasing all motion reproduction once it is identified that a limit will

be encountered would feel strange to the driver, a theory that will be tested later.

In terms of tuning the MPC algorithm has a clear advantage that only the prediction horizon needs to be chosen, and that a single 'long enough' horizon will suit all motion types. The LQR cost function weights and the associated performance trade-off are more easily understood by a non-expert, as are the adaptive cost weights. However, the filter parameters of the adaptive and classical algorithms are not as intuitively tuned by a non-expert.

4. PROPOSED SIMULATOR STUDY

The logical next step in the research is to evaluate the cueing algorithms using tests on a driving simulator with human test subjects. As well as verifying and exploring the relative merits of each algorithm as claimed in the literature, this study will also provide useful data and experience for use later in the research – both in terms of which control strategies to potentially use as part of a new algorithm and of how to carry out successful comparisons of cueing algorithms on the simulator.

The study will involve around 30 test subjects in three age groups: 18-30, 30-50 and 50+. It is intended that there will be approximately equal numbers in each age group, and that within each age group both genders and a range of driving experience levels will be represented. The test subjects will be asked to perform a series of standard manoeuvres – a double lane change, a constant radius turn at fixed speed, and a decreasing radius turn at fixed speed. This set of manoeuvres has been chosen to provide transient, steady-state and limit handling scenarios in which to evaluate the algorithms. Each manoeuvre will be repeated for the different cueing algorithms, the order of the algorithms varied each time to ensure a blind test.

Two types of algorithm evaluation are proposed; evaluation based on the opinions of the test subjects, and based on the subjects performance in controlling the vehicle. Driver opinions will be collected by asking subjects to rate the motion after each test run, probably relative to a baseline motion algorithm (most likely the classical algorithm). Assuming a baseline condition is used, subjects will be asked to rate each algorithm on a scale of -5 to +5 (negative being worse than baseline, positive better) in two areas – ease of vehicle control and overall quality of motion. The results from all subjects will then be analysed using appropriate statistical techniques.

Subject performance in controlling the vehicle will be evaluated based on recorded vehicle data. Path deviation will indicate how well the drivers were able to control the vehicle, and steer angle data will provide information about the control effort required for the manoeuvres – the theory being that better quality motion, i.e. better feedback about the vehicle state, will allow better control with lower control effort.

5. CONCLUSION

The four main algorithm architectures presented in the published literature were discussed and some simulation results presented. The classical, adaptive and LQR algorithms all have a high-pass response, and have a tendency to produce false cues; the adaptive algorithm acts to reduce the magnitude of these false cues. Of these three, the LQR algorithm is most easily tuned by a non-expert, the cost function weights representing a trade-off between motion fidelity and platform excursion. The MPC algorithm behaves very differently to the other three and is much easier to tune, however it is suggested that the resultant motion would feel unusual to the driver. These results are a useful first step in the work but it is difficult to draw any firm conclusions, such is the subjective nature of motion perception.

The planned simulator tests, with a range of test subjects performing several different manoeuvres for each algorithm, will hopefully provide more insight into the relative merits of each motion cueing algorithm, and the results will influence the development of the new algorithm.

REFERENCES

- [1] Schmidt, S.F. and Conrad, B. "Motion Drive Signals for Piloted Flight Simulators", NASA Report CR-1601, 1970
- [2] Reid, L.D. and Nahon, M.A. "Flight Simulation Motion-Base Drive Algorithms Part 1 – Developing and Testing the Equations", UTIAS Report 296, 1985
- [3] Reid, L.D. and Nahon, M.A. "Flight Simulation Motion-Base Drive Algorithms Part 2 – Selecting the System Parameters", UTIAS Report 307, 1986
- [4] Reid, L.D. and Nahon, M.A. "Flight Simulation Motion-Base Drive Algorithms Part 3 – Pilot Evaluations", UTIAS Report 319, 1986
- [5] Berger, D.R., Schulte-Pelkum, J. and Bühlhoff, H.H. "Simulating believable forward accelerations on a Stewart motion platform", Max Planck Institute for Biological Cybernetics Technical Report No. 159, 2007
- [6] Nahon, M.A., Reid, L.D. and Kirdeikis, J. "Adaptive Simulator Motion Software with Supervisory Control", J. Guidance, Control and Dynamics, v15n2, 1992, pp 376-383
- [7] Zacharias, G.L. "Motion Cue Models for Pilot-Vehicle Analysis", Aerospace Medical Research Laboratory Report AMRL-TR-78-2, 1978
- [8] Young, L.R. "The Current Status of Vestibular System Models", Automatica v5, 1969, pp 369-383
- [9] Sivan, R., Ish-Shalom, J. and Huang, J.K. "An Optimal Control Approach to the Design of Moving Flight Simulators", IEEE Trans. on System, Man & Cybernetics, v12n6, 1982, pp 818-827
- [10] Dagdelen, M., Reymond, G., Kemeny, A., Bordier,

M. and Maizi, N. "Model-based Predictive Motion Cueing Strategy for Vehicle Driving Simulators", *Control Engineering Practice*, v17n9, 2009, pp 995-1003