AN ALGORITHM FOR FALSE CUE REDUCTION AND PREPOSITIONING

IG Salisbury [1], DJN Limebeer [1]

(1) : University of Oxford, Department of Engineering Science,
Parks Road, Oxford, UK, OX1 3BN
Phone: 018652 {83172,73810}
E-mail: {ingrid.salisbury,david.limebeer}@eng.ox.ac.uk

Abstract – This paper studies the utility of numerical optimal control (NOC) in the driving simulator motion cueing problem. In this way it is possible to examine how workspace limits affect the cueing signal. A comparative study against Linear Quadratic Gaussian (LQG) and model predictive control (MPC) algorithms shows that false cues are a result not only of platform physical limitations, but also the methods themselves. A scheme is then devised whereby miscues can be manipulated to achieve small acceleration signals that are independent of the onset cueing. The concept is initially tested in a numerical optimal control framework, and having improved the miscue performance, a similar approach is applied to the other algorithms. These new methods produce attenuated miscues, facilitate stronger onset cueing and make better use of the workspace.

Key words: Motion cueing, numerical optimal control

1. Introduction

The critical limitations of all driving simulators are the bandwidth and workspace constraints of the motion platform. Moving platforms simulators were introduced to stimulate the drivers' inertial sensors (vestibular system), thus enhancing the realism of the driving experience and providing a better understanding of the vehicle's behaviour during braking and cornering.

The exact replication of the vehicle's accelerations is prevented by the motion platform constraints. Hence, motion cueing algorithms were developed to determine a simulator reference that approximates the acceleration and thereby improves the simulation.

There are various motion cueing design techniques, the oldest being the classical frequency-shaping algorithm. In this approach the vehicle accelerations are high-pass filtered to generate simulator references. From a workspace perspective the lower frequency motions that correspond to large displacements are removed. With appropriate filter parameter choices, the platform excursions will remain within the limits, although this is only achieved indirectly. From a driver's perspective the initial changes in acceleration, such as at the onset of braking, are important for understanding the vehicle behaviour. The lack of sustained cues later in a manoeuvre is relatively less critical.

Another widely used algorithm is based on LQG theory and includes a model of the human vestibular system. The resulting high-pass filter is designed to minimise the difference between the acceleration sensed by a car driver and driver in the simulator. A more recent approach uses MPC to determine the simulator input reference, while simultaneously reducing perceived acceleration error and directly respecting the workspace constraints.

All of these established approaches have common features: an onset cue, an acceleration washing and a false cue (acceleration in the wrong direction) that prevents the platform hitting the workspace limits. While broadly similar, the methods differ in relation to the detailed production of the above behaviours. An aim of this paper is to determine how the resulting cueing signal is affected by the algorithm choice and platform constraints. Additionally, there should be an identification of the limitations caused by each of these two factors.

In order to compare these methods, a baseline system that determines the 'best achievable' cues for a given platform, is needed. To this end, numerical optimal control is used to determine an open-loop acceleration input for the simulator over a typical driving scenario. This system is not constrained to be linear or causal, and from the result, it is possible to understand the relationship between the platform constraints and the achievable cues/miscues. In the NOC formulation the workspace constraints are recognised explicitly. The NOC results can then be used as a basis for comparison with the other methods and through this, the underlying limitations of each approach can be identified and understood.

This should provide clarity with regards to the available room for improvement and provide insight into how better cueing can be achieved.

These ideas are then used to develop a new strategy that aims to improve the false cue characteristics of more traditional algorithms. An initial study is conducted in the NOC framework, and given the promising results, a similar approach is implemented for the MPC and LQG cases.

The paper is laid out as follows: In Section 2 the various cueing techniques are described. In Section 3 the results of the non-linear optimal control are examined and then compared with the other algorithms. In Section 4 the new technique is described for each method with comparative results provided.

2. Problem Background

The driving simulator involved in this research is used in Formula 1 racing applications. Race driving is characterised by large-magnitude accelerations that cannot be reproduced or sustained in a confined simulator environment. As a result, the motion cueing problem is challenging and critical, since only a small portion of the actual acceleration is supplied to the driver, and the quality of the simulation is heavily reliant on how this is determined.

A commonly employed technique in road car and flight simulators is tilt-coordination. This uses pitch and roll angles to provide a feeling of sustained acceleration in lateral and longitudinal directions. This, however, only applies when the platform can be rotated sufficiently slowly. A second feature of race driving is the fast dynamics, which render this technique useless, further increasing the difficultly of the problem.

Despite the numerous complications associated with the closed-circuit racing context, there are also some advantages. For a given track, the accelerations on each lap are similar, since the drivers tend to be consistent in terms of their driving. This results in a high level of predictability, a characteristic that lends itself to prepositioning [Wei1]. This technique allows the platform to be moved towards an extreme of the workspace (instead of washing out to the centre) in preparation for the next manoeuvre (e.g. braking or strong lateral acceleration); this effectively doubles the available workspace.

Finally, the test drivers themselves are highly skilled, and will tolerate a lack of realism. However, there are cues they need to understand the car behaviour, and it is important that these are delivered.

2.1. Simulator Platform

A conventional 6 degrees-of-freedom (DOF) hexapod is used in this application. It is moved by changing the length of the legs, and the physical constraints of the system (acceleration, velocity and displacement) arise from the specifications of these actuators.

In motion cueing, it is necessary to examine whether an acceleration trajectory demand exceeds the platform's capability. To achieve this, movements in an inertial reference frame need to be transformed into corresponding changes in the leg lengths. The development of these kinematic equations can be found in [Hel1], and the resulting hexapod model is used in the algorithms.

3. Motion Cueing Algorithms

3.1. Linear Quadratic Gauss

The application of LQG theory to the design of motion cueing filters was developed by [Siv1], [Rei1] and [Tel1]. The details of this method are described in those papers, and consequently only a brief summary is provided here.

The human vestibular system can be modelled as a transfer function that takes as input the experienced accelerations, and produces as output the driver-perceived accelerations. In motion cueing the aim is to make the difference between the acceleration sensed by the simulator driver and that sensed by a driver in a real-car as small as possible. However, since the simulator has physical limits, only a fraction of the car's acceleration can be reproduced. A linear filter, *F(s)* is used to process the vehicle acceleration to produce a simulator motion demand. The LQG approach computes the optimal *F(s)* such that the difference between the output of the real car and simulator drivers' vestibular models is minimised. The LQG cost function is defined as follows:

$$J=E\int_{t_0}^{t_f}(a_s^T R_1 a_s+v_s^T R_2 v_s+s_s^T R_3 s_s+e^T R_4 e)dt, \qquad (1)$$

where a_s, v_s and s_s are simulator acceleration, velocity and displacement, respectively, and e is the error in the perceived acceleration. The inclusion of these four terms, requires the LQG to minimise both simulator motion and vestibular error simultaneously. By tuning the weights in the cost function (using trial-and-error), a compromise between these contradictory aims can be found such that the simulator stays within the workspace bounds.

3.2. Model Predictive Control

The use of MPC in motion cueing is described in detail in [Gar1],[Bas1],[Dag1], and, as before, only the pertinent points are provided here.

MPC involves the formulation of an optimisation problem, with constraints, that is then converted into a quadratic programming (QP) problem and solved using any number of techniques.

The core of any optimisation problem is the cost function, which is given in (2). The input *u(t)* is the simulator acceleration demand, and this is to be determined by the solver. The vector *y(t)* is defined in (4), and *R* and *Q* are weighting terms.

$$\min_{u(t)} y(t)^T R y(t) + u(t)^T Q u(t) \qquad (2)$$

Subject to

$$x(t+i+1) = Ax(t+i) + Bu(t+i) \quad i = 0..H_P \qquad (3)$$

$$y(t+i) = x(t+i) - x_{ref}(t+i) \quad i = 0..H_P \qquad (4)$$

$$x_{ref}(t+i) = x_{ref}(t) \quad i = 0..H_P \qquad (5)$$

$$u_{MIN} \leq u(t+i) \leq u_{MAX} \quad i = 0..H_U-1 \qquad (6)$$

$$v_{MIN} \leq v_{PLAT}(t+i) \leq v_{MAX} \quad i = 0..H_P \qquad (7)$$

$$s_{MIN} \leq s_{PLAT}(t+i) \leq s_{MAX} \quad i = 0..H_P \qquad (8)$$

$$u(t+i) = 0 \quad i = H_U..H_P \qquad (9)$$

The first constraint (3) represents the system dynamics, and contains the vestibular system and linearised platform models. From the input (platform acceleration: *u(t)*) the system states (*x(t)*) can be computed. These include the sensed acceleration in the simulator, the vestibular states and the velocity and displacement of the platform actuators.

The vector *y(t)* is representative of the output to be minimised and is calculated as the difference between the system states and a given reference ($x_{ref}(t)$). This reference contains the vestibular states, accelerations sensed in the real car, and components that correspond to the actuator neutral velocity (i.e. *0 m/s*) and length.

A non-zero output therefore occurs when the sensed accelerations do not match, and the platform moves, thus, similar to the LQG approach, both perception error and platform motion are penalised.

A reference acceleration signal is required over the control horizon (H_P). Either a prediction of the future acceleration is needed, or, this can be assumed constant, as given in (5).

Constraints (6), (7) and (8) are a direct result of the actuator acceleration, velocity and displacement limits.

Finally, the input *u(t)* is computed over the control horizon (H_U, where $H_U \leq H_P$) and is then assumed to be zero over the rest of the prediction horizon (9).

The optimisation problem is solved at every time step, but only the input computed for the current step (*i=0*) is applied to the simulator.

3.3. Numerical Optimal Control

3.3.1. Theory

An optimal control calculation computes the state and control vectors associated with a system in order to minimise a performance index [Bet1]. The cost can be expressed in Bolza form, and is given by:

$$J = \Phi(t_0, x(t_0), t_f, x(t_f)) + \int_{t_0}^{t_f} l(t, x(t), u(t))dt. \qquad (10)$$

The system constraints are described by:

$$\begin{cases} \frac{dx}{dt} - f(t,x(t),u(t),p) = 0 \\ g(t,x(t),u(t),p) = 0 \\ h(t,x(t),u(t),p) \leq 0 \\ g_b(x(t_0),x(t_f),u(t_0),u(t_f),p) = 0 \end{cases} \qquad (11)$$

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

Direct methods are used to convert the infinite-dimensional optimal control problem into a finite-dimensional optimisation problem with algebraic constraints; a nonlinear programming problem (NLP). In this application GPOPS II is used to solve the NOC problem.

3.3.2. Application to Motion Cueing

The motion cueing optimal control problem is defined with the following performance index and constraints:

$$J = \int_{t_0}^{t_f} (\hat{a}_{ref}(t) - \hat{a}_{sim}(t))^2 dt \qquad (12)$$

Subject to:

$$\dot{x} = Ax(t) + Bu(t) \qquad (13)$$

$$a_{MIN} \le u(t) \le a_{MAX} \qquad (14)$$

$$v_{MIN} \le v_{act}(t) \le v_{MAX} \qquad (15)$$

$$s_{MIN} \le s_{act}(t) \le s_{MAX} \qquad (16)$$

The control input, *u(t)*, is the simulator acceleration and is constrained by the actuator capabilities (14). This acceleration demand is used to compute the system states, *x(t)*, from the system dynamics described in (13), which include the vestibular model, and the nonlinear hexapod kinematic equations. The actuator velocity and displacement are therefore components of the state and can be limited in (15) and (16). The acceleration sensed in the simulator $\hat{a}_{sim}(t)$, required in cost function, is also a system state.

Finally, a reference which is the sensed acceleration in a real car, $\hat{a}_{ref}(t)$, is required over the optimisation interval (for example a lap of a race track). The optimal control calculation computes the simulator motion demand over the interval in order to minimise the integral of the squared perception error.

4. Comparison of Cueing Techniques

Each of the above algorithms is implemented for the longitudinal freedom during a typical braking manoeuvre. The results from these are discussed, beginning with the numerical optimal control as a basis for comparison.

4.1. Numerical Optimal Control

The simulator acceleration demand computed by the NOC solver is shown in Figure 1 together with the corresponding platform velocity and displacement. The platform has been constrained to begin the manoeuvre at the front of the workspace and end it at the rear; effectively prepositioning it. The acceleration signal will produce the minimum error as defined in the cost function, and it is in no way constrained to be a linearly filtered version of the reference.

This result can be used to draw conclusions about the effects of the workspace on the cueing fidelity. Firstly, by observing the velocity signal, it can be seen that this, and not the displacement, limits the duration of the acceleration signal; the velocity limit is reached before the workspace limit. Secondly, the

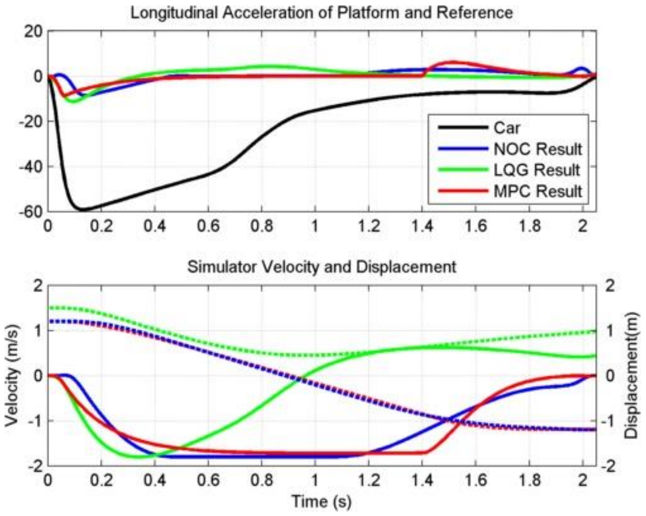

Figure 1 Simulator acceleration, velocity and position (dotted lines) demands computed by the three algorithms.

miscue to slow the platform is delayed until the braking acceleration has decreased in magnitude. This will result in minimal acceleration error and the platform will not violate the constraints.

4.2. Linear Quadratic Gauss

The results of the LQG approach vary depending on the values of the weighting parameters. Adjusting these values is used not only to constrain the platform to move within the workspace, but also to shape the acceleration signal in terms of strength and duration of onset cue. In order to compare the result with the NOC, the LQG is tuned to have a similar cue to the optimal control; the results are shown in Figure 1.

During the onset cue, the velocity once again approaches the constraint. It does not reach the limit however, because the LQG method is difficult to tune to use the full workspace exactly. After numerous iterations it is possible to design filters that come close to the workspace boundary.

In the position demand, some prepositioning was assumed, and the platform is placed initially at the front of the workspace.

The primary difference between the results of the two methods lies in the shape, and timing of the miscue. The LQG miscue immediately follows the onset cue and is of a higher magnitude than the NOC. The shape and duration of this false cue are a direct result of using a linear washout filter. This miscue cannot be adjusted without changing the shape of the onset cue as well. The stronger miscue results in the platform velocity being quickly reduced to zero, and the platform not utilising the full workspace to slow down.

4.3. Model Predictive Control

The MPC cueing problem has more parameters that require adjusting in the tuning process, since not only the weighting vectors can change, but also the length of the control and prediction horizons.

Figure 1 shows the results of the MPC with a control horizon equal to a prediction horizon of 20ms. Since the platform bounds are included as limits in the problem, the workspace is fully utilised.

The results of this are similar to the NOC solution, which is to be expected. If it is assumed that the reference signal is known for the duration of the braking manoeuvre and the control and prediction horizons are set to the whole length of this, the two problems become equivalent (when the plant is linear). The MPC approach differs from the NOC because the look ahead is limited, and the reference is assumed constant in the future.

The key factor in the MPC framework is that the time over which the input is determined can be less than the time over which the system dynamics are solved. So, in an extreme example, if the control is only computed at one time step, and the prediction horizon is 10 steps, then the only way the QP solver can minimise the cost is by changing the input at the first step. During braking, the perception error is minimised by demanding the maximum allowable acceleration at the first time step. Changing the length of the prediction horizon will affect how soon the MPC will determine that the velocity constraints are going to be violated, and this affects how sharply the acceleration is washed out. By adjusting the two horizon lengths the onset cue can be tuned to be both strong and smooth.

Once the platform has reached its maximum velocity, that constraint becomes active and there is no means to reduce the error. At this point, slowing the platform will only increase the perception error, so it is undesirable.

When the platform approaches the edge of the workspace the displacement constraint also becomes active and the platform has to be slowed down. The length of the prediction horizon determines how soon the platform starts decelerating, and the length of the control horizon affects how strong the miscue is. If the control horizon is short with respect to the prediction horizon, then naturally a strong miscue is needed since the input is only non-zero for a fraction of the prediction horizon.

Unfortunately, a longer prediction horizon that will improve the miscues will necessarily also affect the onset cue – making it weaker. Thus, a compromise is needed to produce a good onset cue and an acceptable miscue.

4.4. Remarks

The comparative study of these 3 motion cueing algorithms has made certain things apparent. Firstly, the shape and strength of the miscue is dominated by the platform velocity constraint. This physical limit cannot be avoided and so the onset can be made stronger, but then must be shorter to prevent the platform velocity saturation.

Secondly, the miscues that slow the platform as it approaches the limits are a by-product of the technique employed. Neither LQG nor MPC achieve as smaller miscue as the NOC method. This highlights the need for an alternative miscuing strategy, since it is physically possible to improve them.

Considering these two issues in turn: during onset cueing the aim is to reduce the perception error, however, during miscuing the platform acceleration should be minimised. Ideally, these two problems should be addressed separately so that they can be tuned and adjusted independently. The next section of this work examines the implementation of this idea in the various strategies to determine if, through this, the false cues can be noticeably improved.

5. New Miscuing Strategy

5.1. Numerical Optimal Control

5.1.1. Implementation

The NOC method is the easiest framework in which to implement changes in the cueing strategy, and analyse what improvements can be made that are physically possible. Consequently, the new miscuing idea is applied and tested first in this approach.

A new performance index is given in (17). This function is time-varying and non-linear.

$$J = \int_{t_0}^{t_f} (\beta e^2 w(t) + a_{sim}^2 (1 - w(t)))dt \tag{17}$$

$$w(t) = \frac{1}{2}(1 + atan(- k(t - t_0))) \tag{18}$$

The function *w(t)* is a smooth switch that changes from 1 to 0 at time t_0. The sharpness of the switch is controlled using the parameter *k*. The term *(1-w(t))* is the inverted switch, i.e. it starts at zero and switches to 1 at t_0. The first term in the cost function contains the product of the perception error and the switch, and the second term, the product of the acceleration and the inverted switch. By defining the cost as such, during the first t_0 seconds of the manoeuvre, only the perception error will be minimised, but after t_0 seconds, the perception error will be ignored and the platform acceleration minimised. The weighting term β is included to place more importance on the onset cueing than the miscuing; the resulting miscue should not be considered when designing the onset cue – only the velocity constraint should affect how the cue is shaped.

5.1.2. Results

The results of the braking manoeuvre applied to the newly defined optimal control are shown in Figure 5 below. The onset cue was designed to be strong, reaching the maximum platform acceleration, and the switching time was chosen to be after the velocity limit was saturated.

The improvement in the acceleration miscue is apparent. The acceleration signal is very smooth and the miscue achieves a maximum value of $2m/s^2$. This false cue can be further reduced by removing the constraint that the platform must finish at the rear of the workspace; however, the prepositioning characteristic is more desirable than the slight improvement.

Based on the success of this, the idea was then applied in the MPC and LQG approaches.

5.2. Model Predictive Control

5.2.1. Implementation

In order to improve the false cueing characteristics in the MPC framework the problem needs to be reformulated. At first it seems sensible to make adjustments to the MPC cost function to minimise only acceleration not perception error during miscuing.

However, as mentioned previously, the choice of horizons affects the shape, magnitude and duration of the miscue. As also noted, the horizon cannot be tuned for best miscue without changing the onset cue. Unless, at different times in the braking manoeuvre, the horizons are changed.

The principle is to complete the braking onset cue, and then, once the platform has reached the maximum velocity, the horizons are changed and the control input computed for this adjusted problem. This approach has numerous advantages. The MPC problem does not need to be reformulated as the cost function does not change, and it allows the miscue and onset cues to be tuned independently.

5.2.2. Result

Figure 5 shows the result of changing the horizons during the manoeuvre. The onset cue has been tuned for a stronger deceleration and the false cue has been tuned to be small in magnitude. It is worth noting that if the prediction horizon is too long during the miscue phase then the platform will not fully utilise the workspace.

5.3. Linear Quadratic Gauss

5.3.1. Available Miscue Reduction Techniques

In the context of the LQG and classical filtering approaches other techniques have been developed to reduce the miscues. [Rey1] introduced an additional non-linear gain in the classical approach that reduced the magnitude of the miscues. This is powerful in the classical context where washout is performed by a subsequent filter. However, in the LQG method if the false cues are only gain reduced, then the platform will not stop. This can be remedied by adding an additional washout filter; however, any additional filtering will necessarily affect the onset cues as well. The aim needs to be to not only reduce the gain of the miscues, but also increase the duration and make full use of the available workspace.

5.3.2. Implementation

As it was concluded before, it is not possible to have one filter that is tuned to give both good onset and false cues. There are different requirements during each phase, so the LQG method is now extended to include two filters. The first is optimised to reduce sensation error during the onset cue and the other aims to reduce the acceleration during the false cueing stage. The first filter is designed using the standard LQG approach, and the second is detailed below.

The objective during the miscuing phase is to move the platform to a target position (in preparation for the next manoeuvre), and finish with zero velocity. This can be achieved with a simple controller that feeds back the displacement error and the velocity. This formulation will drive the position to the given reference and the velocity to zero. A diagram of the controller is given in Figure 3 (this also includes bumpless transfer compensation that is still to be explained). The two error signals are gained and summed to produce the acceleration reference. This is described by the Eq. 19, where a_{out}, v_{out} and s_{out} are the platform acceleration, velocity and displacement demands respectively; s_{ref} is the desired platform position and K_1 and K_2 are the feedback gains.

$$a_{out} = K_1(s_{out} - s_{ref}) + K_2(v_{out}) \qquad (19)$$

Finally, it is not recommended simply to switch between two filters, because at a given time there is no guarantee that the states are the same, so there is a 'bump' in the output signal during switching. This is a well-documented problem and is overcome by using a bumpless-transfer scheme. The idea of this is to include feedback around each of the filters that causes the inactive filter output to track that of the active filter. So, when switching occurs, both filters have the same output and no bump appears. The complete switching system is shown in Figure 2, and the details of the two bumpless transfer filters are described below.

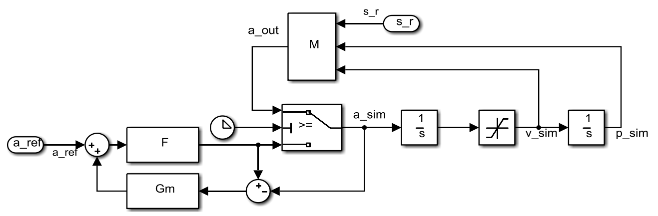

Figure 2 Complete scheme with LQG filter (F), bumpless transfer filter (Gm), and controller M(s) that contains bumpless transfer feedback.

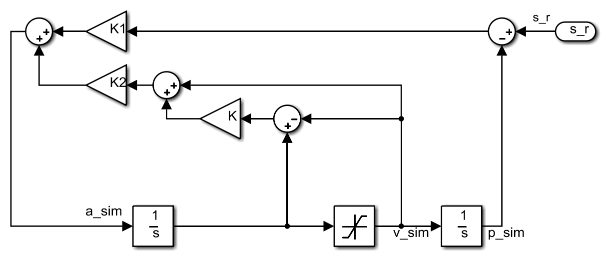

Figure 3 Diagram of controller to slow and preposition platform, with bumpless transfer compensator (K).

The first problem is to ensure that the miscue controller tracks the LQG filter output. The inputs to the controller are velocity and position, and the output is acceleration, which can be integrated to give velocity. It is therefore required that the velocity signal of the miscue controller be the same as the platform velocity when the LQG filter is active. This is achieved using a simple form of bumpless-transfer [Edw1], the implementation of which is shown in Figure 3. The difference between the controller and simulator velocities is gained and added to the feedback loop, adjusting Eq. 19 to:

$$a_{out} = K_1(s_{out} - s_{ref}) + K_2(v_{out} + K(v_{out} - v_{LQG})) \qquad (20)$$

This mechanism has an additional advantage. The output of the controller may produce velocity demands that exceed the platform limits, by including this additional feedback when the velocity saturates the controller output is forced to match the saturated signal and thus the platform acceleration demand remains sensible.

A similar mechanism is required for the LQG filter so that when it is switched back in there is no bump. The simple gain technique did not produce satisfactory results, so an $\mathcal{H}_\infty$ optimisation framework, as introduced and described by [Edw1], was employed to design the feedback.

The bumpless transfer system is described by Figure 4, where a_{sim} is the simulator acceleration demand, a_{out} is the output acceleration of the LQG filter *F(s)* and a_{ref} is the input acceleration. The aim is to design a filter $G_m(s)$ that minimises the error between the output acceleration and the actual platform acceleration. To formulate a sensible $\mathcal{H}_\infty$ problem it is appropriate to minimise both the error and the output from the $G_m(s)$ filter [Edw1].

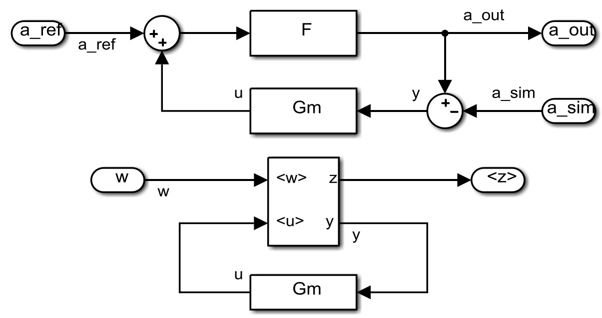

Figure 4 Bumpless transfer feedback filter scheme, and the system recast in generalised regulator form.

This problem can be recast in a generalised regulator format, where:

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \lambda(a_{out} - a_{sim}) \\ \beta u \end{bmatrix} \qquad (21)$$

$$y = a_{out} - a_{sim} \qquad (22)$$

And w and u are given by :

$$w = \begin{bmatrix} a_{ref} \\ a_{sim} \end{bmatrix} \qquad (23)$$

$$u = G_m(a_{out} - a_{sim}) \qquad (24)$$

The resulting plant matrix P is given as:

$$P = \left[\begin{array}{cc|c} \lambda F(s) & -\lambda & \lambda F(s) \\ 0 & 0 & \beta \\ \hline F(s) & -I & F(s) \end{array}\right] \qquad (25)$$

From this formulation the optimal filter $G_m(s)$ can be found using standard robust control algorithms. The parameters λ and β are used to weight the relative importance of the error and the magnitude of the output from the filter.

5.3.3. Results

The results from this approach are given in Figure 5, and show an improved miscue, and consequently, workspace usage.

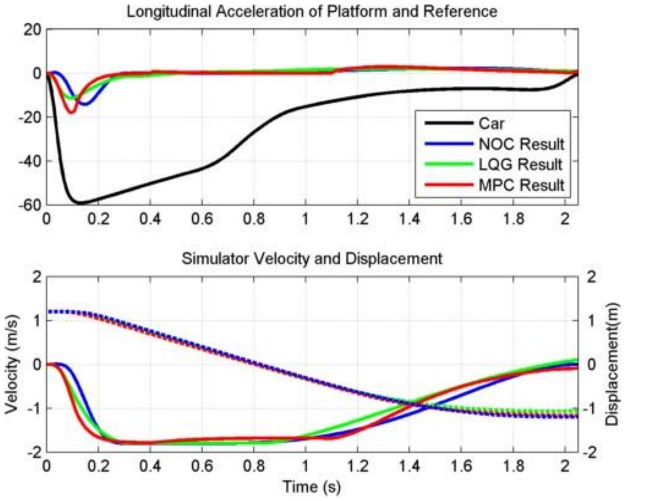

Figure 5 Simulator acceleration, velocity and position (dotted lines) demand calculated using the 3 methods with improved miscues.

6. Future Work

These techniques, having achieved good results in 1DOF, now need to be extended to multiple degrees of freedom.

In the LQG and MPC cases, this needs to be applied in real-time. This requires additional pre-processing of the track to estimate when the respective filters or horizons should be used.

Finally, once implemented on the simulator the driver response will be the ultimate assessment of the improvement.

7. Conclusion

The use of numerical optimal control as a baseline system proved effective in determining the possible areas of improvement in the motion cueing. It was shown that although false cues are physically necessary, the magnitude and duration are a side-effect of tuning for good onset cues, and they can be improved while still respecting the platform physical limits.

A need was identified for methods to separate the onset and false cue phases so that they can be tuned independently, rather than at the expense of each other.

Different means of achieving this were suggested and tested for each algorithm. The results showed a marked improvement in the false cues, and allowed the desired cues to be tuned for a stronger onset.

8. References

[Bas1] Baseggio M. et al. An MPC approach to the design of motion cueing algorithms for driving simulators. In 14th International IEEE Conference on Intelligent Transportation Systems Washington, DC, USA. 2011.

[Bet1] Betts J.T. Practical Methods for Optimal Control Using Nonlinear Programming. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2001.

[Dag1] Dagdelen M. et al. Model-based predictive motion cueing strategies for vehicle driving simulators. In Journal of Control Engineering Practice. 17:955-1003, 2009.

[Edw1] Edwards C., Postlethwaite I. Anti-windup and bumpless transfer schemes. In Automatica, 34:2, 199-210, 1998.

[Gar1] Garrett N.J.I., Best M.C. Model predictive driving simulator motion cueing algorithm with actuator-based constraints, Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility, 51:8, 1151-1172, 2013.

[Hel1] Helinski A.L. Dynamic and kinematic study of a Stewart platform using Newton-Euler techniques. Technical report, U.S. Army Tank-Automotive Command, 1990.

[Rei1] Reid L.D., Nahon M.A. Flight simulation motion-base drive algorithm: Part 1 - developing and testing the equations. Technical Report UTIAS Report No. 296, Institute for Aerospace Studies, University of Toronto, Toronto, Canada, 1985.

[Rey1] Reymond G., Kemeny A. Motion Cueing in the Renault Driving Simulator, Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility, 34:4, 249-259, 2000.

[Siv1] Sivan R., Ish-Shalom J., Huang J. An optimal control approach to the design of moving flight simulators. In IEEE Transactions on Systems, Man and Cybernetics. Vol:12, Issue:6:818-827, 1982.

[Tel1] Telban R.J., Cardullo F.M. Motion cueing algorithm development: Human-centred linear and non-linear approaches. Technical report, State University of New York, 2005.

[Wei1] Weiss, C. Control of a Dynamic Driving Simulator: Time-Variant Motion Cueing Algorithms and Prepositioning. Diploma Thesis. Deutsches Zentrum für Luft und Raufahrt. 2006