A Model-based Motion Cueing strategy for compact driving simulation platforms

Alessandro Beghi [1], Mattia Bruschetta [1], Fabio Maran [1], Diego Minen [2]
(1) University of Padova, Department of Information Engineering Via Gradenigo 6/b, 35131, Padova, Italy,
E-mail : {alessandro.beghi, mattia.bruschetta, fabio.maran}@dei.unipd.it
(2) VI-Grade Italy, Via l'Aquila 1c 33010, Tavagnacco, Udine, Italy, E-mail : diego.minen@vi-grade.com

Abstract *– Driving simulators are widely used in different applications: driver training, vehicle development, and medical studies. To fully exploit the potential of such devices, it is crucial to develop motion control strategies that generate realistic driving feelings. This has to be achieved while keeping the platform within its limited operation space. Such strategies are called motion cueing algorithms. In this paper a particular implementation of a motion cueing algorithm is described, based on Model Predictive Control technique. A distinctive feature of such approach is that it exploits a detailed model of the human vestibular system, and consequently differs from standard motion cueing strategies based on washout filters. The algorithm has been evaluated experimentally on a small-size, innovative platform, by performing tests with professional drivers. Results show that the MPC-based motion cueing algorithm allows to effectively handle the platform working area and to devise simple and intuitive tuning procedures.*

Key words: *Motion Cueing, Model Predictive Control, Vestibular System, Dynamic Platform, Optimization.*

Introduction

In recent years, there has been a growing interest in the development of *dynamic* driving simulators with applications in different fields such as racing (e.g., professional driver training, virtual vehicle development and set-up), security control systems (e.g., accident avoidance), medical rehabilitation, and virtual prototyping (providing safe and realistic virtual environment for Hardware-In-the-Loop (HIL) tools). In particular, given the growing number of possible applications, focus has been given to the design of small-size, low-cost platforms.

In a dynamic simulator, it is crucial to faithfully reproduce the driving feelings, so that the driver can fully exploit the virtual experience for the given specific goal. To this aim, the *motion cueing* (MC) strategy, i.e. the algorithm used to transform vehicle accelerations into admissible motion commands to the platform, has a key role. One of the main difficulties in the design of effective motion cueing algorithms is given by the complex nature of the human perception systems, since from a physiological point of view the role and priorities of stimuli of different nature to the overall perception of accelerations is not yet well known.

In most dynamic simulators, motion cueing algorithms are based on the so called "classical" approach **[Nah1]** that basically consists of a sequence of filters (figure 2) combined in order to:

- remove low frequency components of accelerations and velocities obtained from the vehicle dynamic model;
- transfer part of the low frequency translational accelerations to the angular dynamic using a low pass filter (*tilt coordination*);
- limit the platform motions with a further high pass filter to keep the platform in a neutral position. This is commonly called *washout action*.

This simple strategy has seen a wide range of implementations over the years. However, it has some shortcomings:

- being a filtering based approach, it is not possible to guarantee stimuli consistency between the dynamic simulation environment and the real platform movements;
- it cannot explicitly handle hard constraints on the platform movements and accelerations;
- it is not possible to exploit any available information on the driver's behavior in the future;
- the tuning of the algorithm is in general difficult, since it is not easy to give physical interpretation to most of the parameters.

In this paper the experimental application of a MC algorithm based on the Model Predictive Control (MPC) technique for a small size dynamic simulator is described. MPC is a model-based control methodology that allows to effectively handle limits on the working space and to exploit information on future reference signal. The idea of using MPC for MC has been recently proposed in **[Dag1]**, **[Aug1]**, where the motion cueing strategy integrates a model of the human perception systems and takes advantage of predictions of the future trajectory to fully exploit the platform working area. In these early works, the proposed solutions are not suitable for experimental application in real situations. Moreover, they focus on investigating the prediction capabilities rather than on taking advantage of the optimization approach of MPC. A similar approach was used in **[Bas1]** with application to the design of the MC algorithm for an innovative, small size driving simulator platform. The results presented in **[Bas1]** were based on simulations. In the present work, starting from the results of **[Bas1]**, a real time implementation of an MPC-based motion cueing strategy for the same driving simulator platform as in **[Bas1]** is described. Performance of the implemented algorithm is evaluated on the field by professional test drivers. In particular, it is shown that the MPC-based approach gives satisfactory performance even in the case where no prediction of future driver's behavior are available, it allows to effectively handle the platform working area, to limit the presence of those platform movements that are typically associated to driver motion sickness, and to devise simple and intuitive tuning procedures.

The paper is organized as follows. The experimental platform is first described, and the motion cueing design problem is stated. For the sake of readibility and self containedness, basics of MPC control applied to the design of MC algorithms are briefly reviewed. In particular, the performance index, the optimization algorithm to be used to solve the MPC problem and the model of the human vestibular system are described. Operation and tuning of the motion cueing algorithm are then discussed by means of experimental results.

Problem Statement

In Fig. 1 the platform considered in this study is represented. Its peculiarity is in the mechanical structure. By using linear actuators instead of the classic hexapodal structure, it is possible to achieve satisfactory results in physical simulation with a relatively small size hardware, that can fit standard laboratories environments, whereas traditional, large dimensional, hexapodal platform require dedicated hangars.

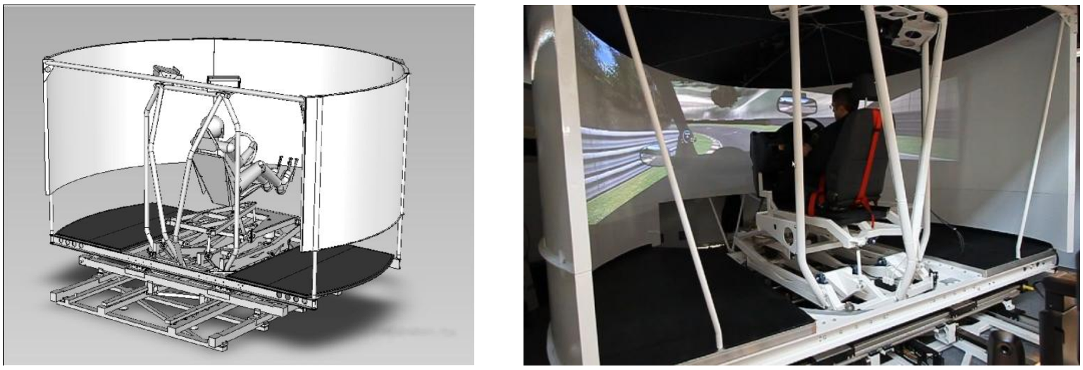


Fig. 1: Experimental platform : schematics and picture.

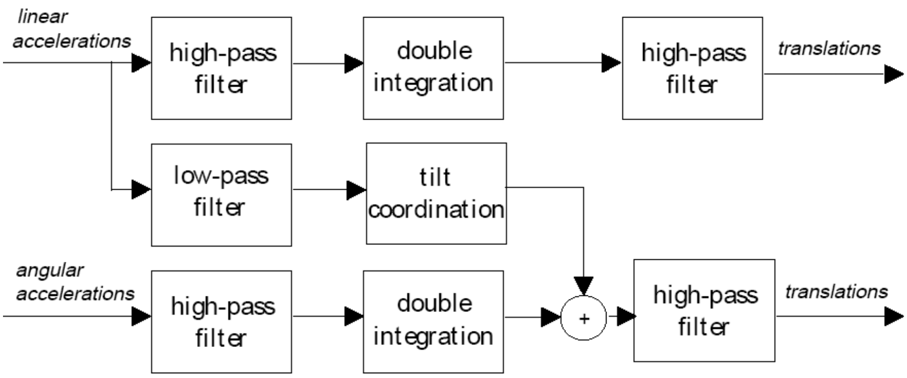


Fig. 2: Classical Motion Cueing (Washout).

The architecture is based on three completely decoupled degrees of freedom (DOFs, longitudinal and lateral axis, and yaw), and three partially coupled DOFs. The simulator kernel, i.e. the vehicle dynamics physical engine, has been developed and extensively tested on the field and provides a highly reliable representation of the real vehicle behavior. The screen covers more than 180 deg and moves in agreement with the platform to guarantee full immersion of the driver in the virtual environment. Finally, force feedback on the steering wheel and the braking system enhances the “driver's feeling” of the vehicle behaviour.

The platform dynamic performance reported in Tab. 1 highlights the limitations of the operational space, with maximal linear excursions of 1 m. This fact makes the role of the MC algorithm crucial.

Table 1. Platform performance.

Range	Position	Velocity	Acceleration
x	1m	1.3m/s	3.3m/s ^{2}
y	1m	1.3m/s	3.6m/s ^{2}
z	0.3m	0.9m/s	4.9m/s ^{2}
Roll	30deg	112deg/s	600deg/s ^{2}
Pitch	24deg	61deg/s	600deg/s ^{2}
Yaw	50deg	61deg/s	240deg/s ^{2}

The MC strategy has to provide the displacement references to the control system of the platform, which is assumed to be able to perfectly track the reference signals, with a fixed time delay. The conceptual scheme of the MC procedure is shown in Fig. 3, and is composed by the following steps:

1. obtain the current useful vehicle states d, i.e. translational acceleration and angular velocities calculated on the driver's eye-point, from the simulation software;
2. obtain the *perceived* acceleration r by filtering d via the vestibular system model, thus generating the reference signal for the MPC algorithm;
3. compute via MPC the displacement signal p passed to the platform control system in order to achieve the desired behavior on the eye-point.

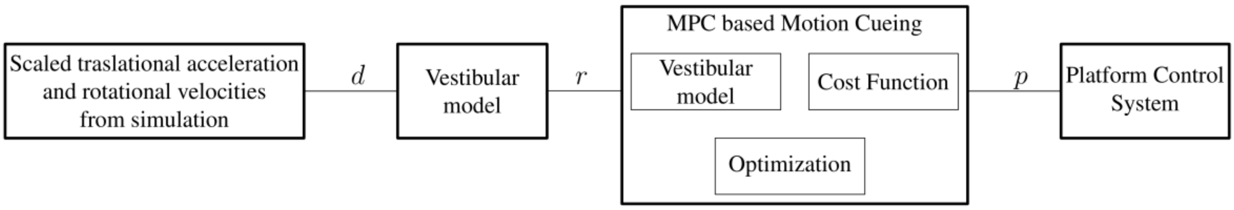


Fig. 3: Scheme of motion cueing strategy.

Model Predictive Control

Model Predictive Control (MPC) is an advanced control technique widely used in industrial applications **[Wan1]**, **[Mac1]** since the 1980s. In recent years, robust and efficient implementations have been developed, as well as software tools in standard computational environments that ease the design of MPC algorithms. The main advantages of MPC can be summarized as follows:

- its underlying idea is simple and intuitive to understand;
- it's the only generic control technique that efficiently deals with constraints;
- it can handle Multi-Input Multi-Output (MIMO) systems without formally increasing the complexity of the problem;
- it can handle non linearities in both the model and the constraints.

MPC Basics

Assume (discrete-time problem) that at time k a reference trajectory $r(t|k)$, $t \geq k$ and a current measure of the output $y(k)$, are available. Note that the current input is *not yet computed*. Now, suppose to have a model of the process to be controlled and that the state of the system (or its estimate) is available. We can therefore predict the future output $y(k+i|k), = 1, \dots, N_p$ corresponding to the input sequence $u(k+i|k), = 0, \dots, N_p - 1$ in a time-window of length N_p, where N_p is the prediction horizon length (Fig. 4). The idea is to compute the input sequence $\hat{u}(k+i|k)$ that minimizes a cost function, e.g. a function of the tracking error

$$\varepsilon(k+i|k) = r(k+i|k) - y(k+i|k),$$

while respecting a set of constraints. The input to be applied at time k is chosen as

$$u(k) = \hat{u}(k|k);$$

at time $k+1$ a new output $y(k+1)$ is measured and the algorithm is iterated applying only the first element of the computed optimal input sequence.

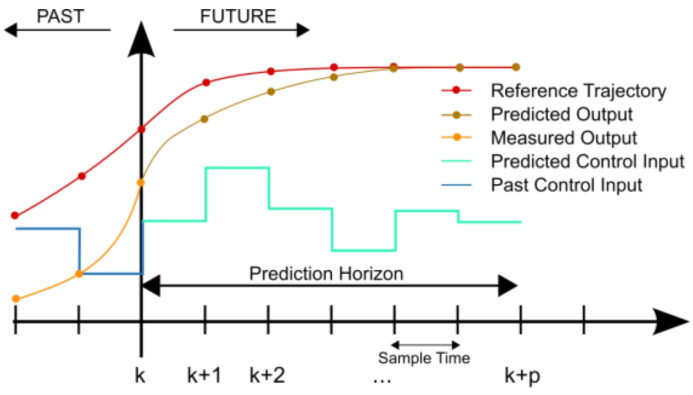


Fig. 4: Representation of MPC principle.

Process Model

In the literature, different implementations of the MPC principle have been proposed, with different model structures. In the application we are considering, the real-time constraints and the MIMO structure of the model are well described by a linear discrete state space model (sampled version of the continuous process) of the form

$$x_m(k+1) = A_m x_m(k) + B_m u(k)$$
$$y(k) = C_m x_m(k).$$

We assume that the input does not have a direct effect on the output (strictly proper system). State-space models are particularly well suited to design state estimators by using well-established tools of statistical filtering theory.

In our case, as proposed by Wang **[Wan1]**, we consider as the element to be optimized the *input difference* $\Delta u(k) = u(k) - u(k-1)$: considering also the state difference $\Delta x_m(k)$, we can write a new state equation

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k).$$

Considering the output difference

$$y(k+1) - y(k) = C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k)$$

and defining a new, augmented state $x(k) := [\Delta x_m(k)^T (k)^T]^T$, we obtain a new model

$$x(k+1) = \begin{bmatrix} A_m & 0 \\ C_m A_m & I \end{bmatrix} x(k) + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k)$$
$$y(k) = \begin{bmatrix} 0 & I \end{bmatrix} x(k)$$

where the control input is $\Delta u(k)$.

Cost function and state estimation

The optimal input sequence $\Delta\hat{u}(k+i|k)$, $i = 0, \dots, N_p - 1$ is computed by minimizing a cost function of the form **[Wan1]**

$$J(t) = \sum_{j=1}^{N_p} \delta(j)(y(t+j \mid t) - r(t+j))^2 + \sum_{j=0}^{N_c-1} \lambda(j) u(t+j)^2 + \sum_{j=0}^{N_c-1} \gamma(j) \Delta u(t+j)^2; \qquad (1)$$

$J(t)$ is quadratic, and takes into account the error between the predicted trajectory and the future reference in the prediction window of size N_p, and the future inputs and *input difference* in the control horizon N_c: in the specific case $N_c < N_p$ (as often done in the MPC framework). In this form, $J(t)$ has to be minimized over $u(t)$ and $\Delta u(t)$. Observe that weights on $\Delta u(t)$ are included in (1). In fact, as will be shown in the next Section, among the system inputs there are *longitudinal accelerations*, that are high-frequency, discontinuous signals. It is therefore convenient to have the possibility to act on their (approximate) derivative, to achieve a certain degree of regularity in the control signal, thus avoiding and excessive stress of the actuators.

The cost function can be rewritten in order to depend only on the input difference ΔU, obtaining the classic, matricial formulation of a *quadratic problem* (QP) **[Wan1]**, that is,

$$J = \frac{1}{2}\Delta U^T H \Delta U + \Delta U^T F.$$

To deal with *constraints*, limitations on the system inputs and outputs can be written in terms of constraints on the input variations ΔU **[Wan1]**

$$A \cdot \Delta U \leq b\,.$$

As a consequence, the QP becomes a *constrained* QP, for which a variety of solving algorithms are available in literature. This is a key step to ensure that the control problem, and consequently the MC algorithm, can be solved in real time.

Optimizer

From the implementation point of view, the Quadratic Programming solver is the core of the MPC algorithm. In the application at hand, there are strict real-time requirements, since fast dynamics (control frequency of 100 Hz) call for small computation times, and this leads to the use of *online QP solvers* which iteratively calculate the result at each sample time, without off-line precalculations.

After having analyzed different solutions, an Active Set method **[Wan2]** has been chosen to deal with the MPC problem described in this paper. The AS algorithm has been proposed by Ferreau, Bock and Diehl in **[Fer1]** and implemented in the tool qpOASES **[qpO1]**. qpOASES freeware C++ implementation provides a ready-to-use package with real-time capabilities that well fit the MPC-based motion cueing algorithm to be implemented in the driving simulator. In particular, the package offers some useful solutions for matching different real-time requirements, as the tunable limitation of the maximum number of active set recalculation per sample step and heuristics to assess the time required to complete the current optimization calculation, while limiting the possibilities of infeasibility if the procedure is stopped before reaching the optimum.

The Vestibular System

The vestibular system is located in the inner ear and is composed by the semicircular canals and the otolith organs. The former sense the angular rotation and the latter linear motion. Accurate mathematical models of the two systems have been derived starting from the '70s for application to MC of flight simulators. Zacharias **[Zac1]**, in a survey written in the 1979, reported most of the results nowadays available. Telban and Cardullo in 2005 **[Hou1]**, **[Tel1]** published a simplified transfer function model with estimates of the corresponding parameter values. For the semicircular canal, the transfer function that can best relate the sensed angular velocity to the acceleration stimulus in a MC control problem is the following:

$$W_{SCC}(s) = \frac{\widehat{\omega}(s)}{\omega(s)} = K_{SCC}\frac{s^2}{(1+T_{SCC,1}s)(1+T_{SCC,2}s)} \qquad (2)$$

The otoliths are described in terms of the following transfer function that relates the sensed response to the specific acceleration stimulus:

$$W_{OTH}(s) = \frac{\hat{a}(s)}{a(s)} = K_{OTH}\frac{1+T_{OTH,1}s}{(1+T_{OTH,2}s)(1+T_{OTH,3}s)} \qquad (3)$$

Tilt coordination

An important component of perception in a dynamic simulator is given by tilt coordination. Otoliths are not capable to discriminate between gravitational and longitudinal forces. Hence, by using a non-zero pitch (roll) angle and without any other visual reference, it is possible to provide the driver in the simulator with a "fake" longitudinal (lateral) acceleration sensation. Such approach goes under the name of *tilt coordination*. Taking into account this effect is crucial to reproduce low frequency behavior with a reduced range working area. In the perception model, because of linearization, tilt coordination is nothing but a further contribution in the otoliths model $W_{OTH}(s)$ due to the pitch angle θ in the longitudinal direction and to the roll angle ϕ in the lateral direction: being $a = [a_x \;\; a_y \;\; a_z]^T$ the acceleration the driver has to be provided with, by using tilt coordination it suffices to generate the *specific acceleration*

$$\tilde{a} = \begin{bmatrix} a_x + g\sin\theta \\ a_y - g\cos\theta\sin\phi \\ a_z - g\cos\theta\cos\phi \end{bmatrix} \approx \begin{bmatrix} a_x + g\theta \\ a_y - g\phi \\ a_z - g \end{bmatrix}, \qquad (4)$$

using the small-angle linearization.

The complete model

In order to use the perception models in the MPC approach, state space realization of $W_{OTH}(s)$ and $W_{SCC}(s)$ are obtained and coupled with the tilt coordination contribution for all the 6 DOFs. The resulting system can be written as

$$\dot{x} = A_{VEST}x + B_{VEST}u$$
$$y = C_{VEST}x + D_{VEST}u$$

where the input u is composed by the three applied longitudinal accelerations and the three angular velocities, i.e $u = [a_x \;\; a_{y_z} \;\; \dot{\theta} \;\; \dot{\phi} \;\; \dot{\psi}]^T$. The overall state vector x is

$$x = [x_{SCC} \;\; x_{OTH} \;\; v_x \;\; p_x \;\; v_y \;\; p_y \;\; v_z \;\; p_z \;\; \theta \;\; \phi \;\; \psi]^T \qquad (5)$$

where the actual angles, positions and velocities are obtained by integration from the inputs u, and x_{OTH} and x_{SCC} are the state variable for the dynamical systems associated with the otoliths and semicircular canals. To impose a set of constraints in a simple manner we choose $y = [\hat{\omega}\ \hat{a}\ v_x\ p_x\ v_{y_y}\ v_z\ p_z\ \theta\ \phi\ \psi]^T$, where $\hat{\omega}$ and $\hat{a}$ are the vectors of perceived angular velocities and longitudinal accelerations along all the DOFs.

Remarks on the operation and tuning of the MC algorithm

As can be seen by analyzing the structure of (1), it is necessary to produce reference trajectories for each of the output variables. This can be done by using the simulation environment, where perceived transactional accelerations and angular rates are generated, and then scaled prior to be used in the MPC algorithm. To keep the platform within its operational limits, as an alternative to the classical washout action, constant zero references for the position of the all six DOFs and for the velocities of the longitudinal ones are used. The weights in (1) are the tuning parameters to achieve a satisfactory trade-off between an accurate reproduction of the perceived in the simulated vehicle and the compliance with the platform working area.

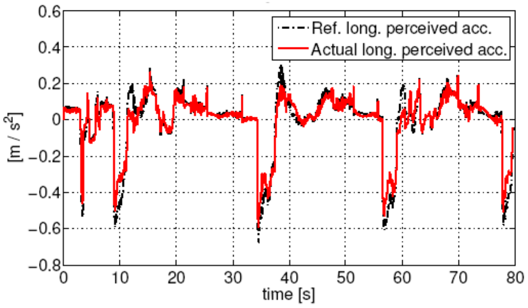
By integrating the gravity effect inside the model as described in the previous Section, the low-frequency tilt coordination correction is automatically achieved giving an important contribution to the tracking of the perceived simulation signals. This fact can be seen as one of the major advantages of the MPC approach to the design of the motion cueing algorithm, namely, a tilt coordination correction can be obtained as the result of a model based optimization procedure.

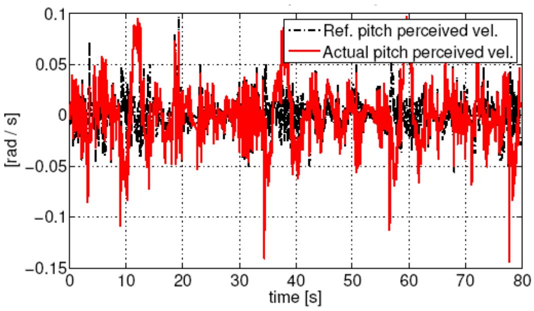
As far as the design of the MPC algorithm is concerned, it would clearly be convenient to make use of the widest prediction/control window possible, if reliable information on the future driver's behavior is available. However, hard real-time constraints and the possible lack of capability to predict the driver's behavior may limit the length of the prediction/control window in practical situations. For these reasons, to design a general purpose Motion Mueing algorithm, it is preferable to first provide an effective algorithm where information about the future driver's behavior is not required and where a small prediction window is used. In the proposed solution, the MPC is designed to keep the reference trajectory constant along the prediction interval, and the length of the window becomes a tunable parameter that can be varied to obtain the desired tracking performance (a common strategy in MPC-based tracking problems for system with slow dynamics). In such conditions the low-frequency component of the desired perception signal is covered by tilt coordination and the high-frequency component is reproduced by using the translational DOFs. The tuning of the algorithm consists in choosing the weights, the length of prediction and control window, and the scaling factors to obtain satisfactory performance of the overall system, in terms of realistic sensations and effective usage of the platform working area. Due to the limited number of pages permitted we do not report the values of the tuning parameters in ditail.

Results

In this Section, some experimental results obtained during a professional driver training session on the platform are reported. The simulated vehicle is a sports-class car, and the virtual test track is a digital version of the Calabogie track. Since the considered model is almost decoupled for all the 6 DOFs, results on longitudinal dynamics only are reported, namely, the longitudinal acceleration and the pitch velocity of the vehicle, that have to be reproduced as faithfully as possible (in terms of driver sensations) by operating the platform with longitudinal and pitch motions.

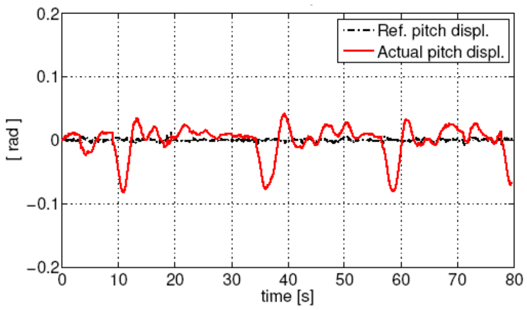
The motion cueing algorithm is tuned so that the platform working area in the longitudinal direction is exploited at best, while avoiding motion sickness due to the tilt coordination correction. In Figure 5, tracking of the perceived accelerations and velocities, longitudinal platform displacement and angular displacement are shown. It can be seen that the platform actually exploits at best its operational area. It is worth noticing that this is achieved *w*ithout any filtering action, as would be required in a classical, washout filters-based approach. Also, the perceived acceleration is tracked almost perfectly. Focusing on the pitch displacement, tilt coordination correction can clearly be seen and has a peak value of 0.08 radians. As a consequence, the angular velocity contains low frequency disturbances. The frequency analysis of pitch and longitudinal contributions to the global perception in the platform, reported in figure 6, shows that the displacement is responsible of the low frequency contributions only. The tuning procedure followed to achieve such results is very intuitive and it is based on the ``natural interpretation'' of the weights in the cost function.

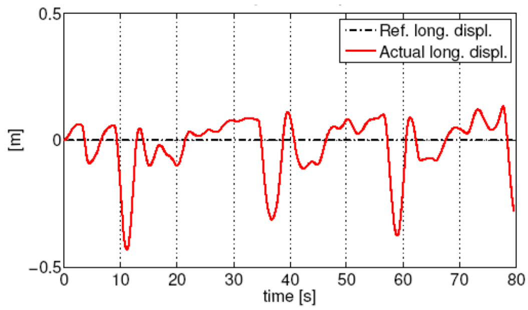




(a) Perceived longitudinal acceleration.

(b) Perceived pitch velocity.





(c) Pitch position.

(d) Longitudinal position.

Fig. 5: Tracking based on MPC approach.

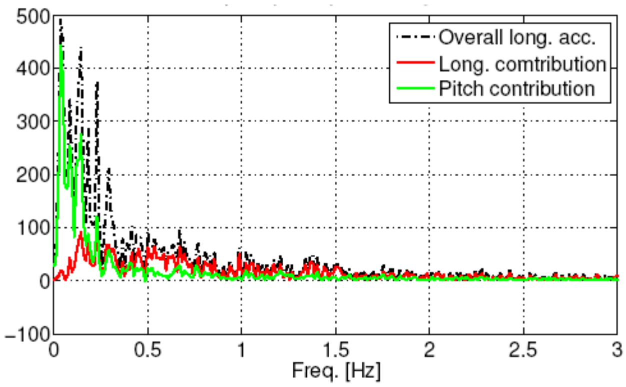
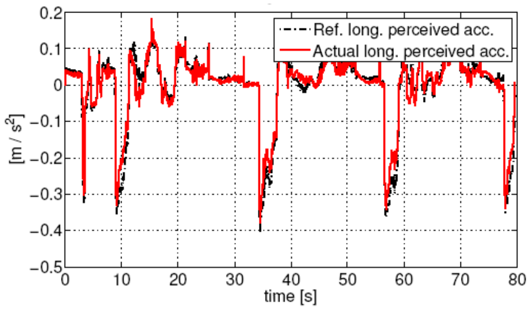
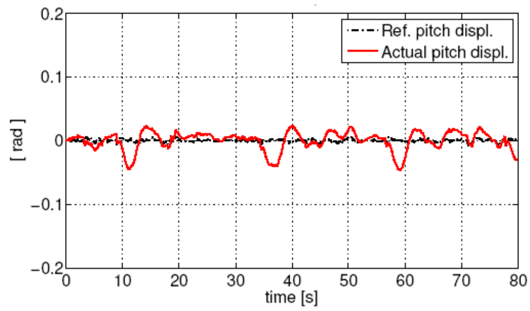


Fig. 6: Frequency comparison of the longitudinal and pitch contribution to the longitudinal acceleration

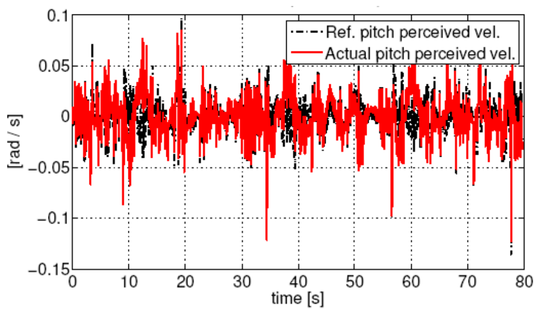
As a further example of the effects of manipulating the weights in (1) can be seen by increasing the values concerning the angular velocity of the platform and by modifying the scaling factor. A reduction of the overall pitch displacement for the platform is achieved, as shown in Figure 7. This operation allows to reduce the tilt coordination effect which is one of the main causes of motion sickness in a dynamic simulator, hence adapting quickly to different drivers' attitude.



(a) Perceived longitudinal acceleration.



(b) Pitch displacements.



(c) Perceived pitch velocity.

Figure 7: Comparison of tracking performance with different exploitation of the platform.

Conclusions

In this paper we describe the experimental design of a MC algorithm for a small size dynamic driving simulator, based on MPC techniques. The proposed algorithm represents a novel approach to motion cueing that completely changes the classic paradigms of washout filters: tilt coordination and working area constraints are handled through an optimization procedure without the employment of any filter. This procedure results to be easily tunable and robust. It is worth noting that, given the high system dimension, although implementing a real time MPC procedure is not a trivial task, our algorithm is working at 100 Hz control frequency. Next step will be using a virtual driver tool to provide reliable driver's behavior prediction and hence extending the proposed algorithm capabilities.

References

[Aug1] Augusto, B.D.C. and Loureiro. Motion Cueing in the Chalmers Driving Simulator: A Model Predictive Control Approach. Master's thesis, Chalmers University of technology, 2009.

[Bas1] M. Baseggio, A. Beghi, M. Bruschetta, F. Maran, and D. Minen, "An MPC approach to the design of motion cueing algorithms for driving simulators," in Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, oct. 2011, pp. 6926-97.

[Dag1] Dagdelen, M. and Reymond, G. and Kemeny, A. and Bordier, M. and Mazi, N. Model-based predictive motion cueing strategy for vehicle driving simulators. *Control Engineering Practice*, 17(9): pp. 995-1003, 2009.

[Fer1] H.J. Ferreau and H.G. Bock and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8): pp.816-830, 2008.

[Hou1] Houck, Jacob A and Telban, Robert J and Cardullo, Frank M. Motion Cueing Algorithm Development: Human-Centered Linear and Nonlinear Approaches. *NASACR*, 213747(May), 2005.

[Mac1] Maciejowski, J.M. *Predictive control: with constraints*. Pearson education, 2002.

[Nah1] Nahon, MA and Reid, LD. ''Simulator Motion-Drive Algorithms: A Designer's Perspective''. *J. Guidance*, 13(2), 1990.

[qpO1] qpOASES website, www.kuleuven.be/optec/software/qpOASES.

[Tel1] Telban, R.J. and Wu, W. and Cardullo, F.M. and Langley Research Center. *Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms*. National Aeronautics and Space Administration, Langley Research Center, 2000.

[Wan1] Wang, L. *Model predictive control system design and implementation using MATLAB*. Springer Verlag, 2009.

[Wan2] Wang, Y. and Boyd, S. Fast model predictive control using online optimization. *Control Systems Technology, IEEE Transactions on*, 18(2): pp. 267-278, 2010.

[Zac1] Zacharias, G.L. Motion cue models for pilot-vehicle analysis. Technical report, bolt beranek and newman inc Cambridge ma control systems dept, 1978.