AN MPC APPROACH TO THE DESIGN OF MOTION CUEING ALGORITHMS FOR A HIGH PERFORMANCE 9 DOFS DRIVING SIMULATOR

Mattia Bruschetta [1], Fabio Maran [1], Alessandro Beghi [1], Diego Minen [2]

(1) : Department of Information Engineering
University of Padova, Italy
{name.surname}@dei.unipd.it

(2) : Vi-Grade s.r.l.
Tavagnacco, Udine, Italy
diego.minen@vi-grade.com

Abstract – Driving simulators (DS) of innovative mechanical structures tailored to specific market needs have been recently designed, due to the increasing diffusion of such devices in many different application fields. The effectiveness of DSs is related to the capability of the motion control strategies of faithfully reproducing the driving feelings, while staying within the operation space. Such strategies are called Motion Cueing Algorithms (MCAs). Their implementation strongly depends on the particular mechanical structure. In this paper, a MCA based on *non-linear* Model Predictive Control (MPC) is considered for a new-concept simulator, which is based on a combination of a hexapod over a flat base moved by a tripod, exhibiting a highly non-linear behaviour. In particular, the main goal is that of increasing performance in terms of yaw DOF exploitation, a crucial one to well reproduce driver feelings that is limited by the architecture of classical platforms. Preliminary results show a full exploitation of the working area, while managing all the limitations given by the structure.

Key words: Motion Cueing, Yaw angle, Hexapodal Simulator, Non-linear Control, Model Predictive Control.

1. Introduction

In the framework of dynamical motion simulation platforms, Motion Cueing Algorithms (MCAs) play a central role. The effectiveness of those devices is strongly related to their capabilities of faithfully reproducing the motion feeling the driver would have inside a real vehicle: a well designed MCA can exploit at best the capabilities of the platform on which they have been implemented, whatever the final aim is (e.g., virtual vehicle prototyping, racing setup and development, rehab.) At the same time, MCAs have to deal with the physical constraints of the specific platform, preventing discontinuities in motion due to the limitations of the actuators that can lead to unfeasible positions, unphysical accelerations and possible damage of the platform. This is commonly known as Washout Action (WA).

Achieving good results is not an easy task, due to the complex nature of the human perception systems: It is not clear yet, from a physiological point of view, the roles and priorities of stimuli of different nature to the overall perception of accelerations and forces. Moreover, human reactions are subjective and experience dependent (e.g. professional pilots are more sensible to some aspects of motion than non professional drivers).

From these considerations, a model-based approach appears to be better suited for this kind of application than the “classical” one based on a simple combination of high-pass/low-pass filters. To this aim, the Model Predictive Control (MPC) paradigm can be used, where model-based, optimal control techniques are employed that make explicit use of constraints, which can include both the physical limitations of the actuators and the human perception system [Wan1]. The use of an appropriate model and cost function makes this approach an efficient, viable solution [Aug1], [Dag1].

In [Bru1, Bru2, Mar1] a MCA based on a MPC technique for a 6 DOFs dynamic platform has been proposed. In that particular setup, given that the degrees of freedom are partially decoupled, the system has been split into four sub-systems to “parallelize” the computation of the optimal solution, thus improving real-time performance. The algorithm has been implemented and widely tested on the testbed machine before been released for use on other platforms. A further improvement has been presented in [Mar2] by introducing a more accurate prediction step, with the exploitation of the repetitive pattern typical of the racing context, together with a decimation strategy to improve the real time performance. Concerning more traditional and complex platform structures, a recent work deals with the problem of exploiting the inverse kinematic model of a classical hexapod, still adopting linear models [Gar1].

One of the main limitations of commercial simulators, for what concerns the automotive field, is the yaw degree of freedom. Satisfactory platform movements along this DOF are crucial but very difficult to achieve. A new concept structure, from now on referred to as DiM (Driver in Motion, see Fig. 1), has been introduced to overcome these difficulties, besides bringing other advantages. From a mechanical point of view, it is composed by a hexapodal structure installed upon a tripodal-actuated plane, able to perform longitudinal, lateral, and yaw displacements: hence the yaw angle is contributed both my the hexapod and the tripod. To develop a motion cueing capable of exploiting at the best all the degrees of freedom of the DiM architecture, a nonlinear, MPC based algorithm is proposed that can be considered a substantial evolution of the algorithm described in [Bru2, Mar1]. In this previous work, in fact, a small platform with decoupled degrees of freedom was considered, whereas the DiM has a relevant coupling between the different degrees of freedom. The two main coupling factors are:

- the continuous, non-linear map between general coordinates and actuators displacement, typical of hexapodal structure.
- the interaction between actuators which is related to the relative position of the tripod over the hexapod. This is a peculiar feature of the DiM.

Starting from an analytical study of the platform motion envelope, this latter factor, specifically along the yaw direction, has been identified as the main obstacle to a wide exploitation of the platform working area. The proposed approach consists in a combination of linear and non-linear real time MPC based motion cueing, capable of avoiding actuators interaction, exploiting a full inverse kinematic model of the platform. Furthermore, the algorithm allows handling in an optimal way the separation of the global yaw displacement between the hexapod and the tripod.

2. Problem statement

The mechanical structure of the simulator consists of a hexapodal structure mounted on a tripod frame, which slides on special air/mag pads on an extremely even and stiff steel surface. In this way, it is possible to achieve satisfactory results in physical simulation with a relatively small size assembly, whereas an equivalent-net-workspace traditional hexapodal platform would require a dedicated hangar. The planar tripod is responsible for longitudinal, lateral, and yaw sliding movements and the hexapod for pitch, roll and vertical ones, while being also used for small longitudinal, lateral, and yaw movements. The redundant DOFs allow to increase the overall bandwidth and to have a large motion envelope while maintaining a limited occupied volume. 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 behaviour [Fre1]. The screen covers over 220deg angle and the projected image moves in proper coordination with the platform to guarantee full immersion of the driver in the virtual environment. Force feedback on the steering wheel and the braking system implements the driver’s feeling of the vehicle behaviour. The platform dynamic performance reported in Table 1 highlight the limitations of the operational space.

The overall system is clearly nonlinear, both in the operational space and the actuators space. When calculating the motion displacements the need for avoiding interference between the actuators must be taken into account, that can result in a dangerous situation for the driver and damage of the device.

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 comprises the following steps:

1. obtain the current vehicle states, i.e. translational acceleration and angular velocities calculated on the driver eye-point, from the simulation software;
2. obtain the “perceived acceleration” by filtering vehicle states via the vestibular system model, thus generating the reference signal for the NMPC algorithm;

3. compute via NMPC the displacement signal to be passed to the platform motion control system in order to achieve the desired behaviour on the eye-point.

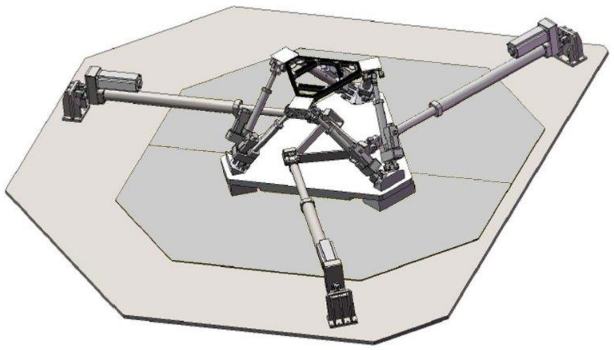

Fig. 1. DiM outline

Table 1. Platform Tripod (t) and Hexapod (h) performance

DOF	Position	Velocity	Acceleration
x_t	±0.8m	1.7m/s	12m/s^2
y_t	±0.75m	1.5m/s	10m/s^2
Yaw, ϕ_t	±25deg	165deg/s	900deg/s^2
x_h	±0.28m	2m/s	25m/s^2
y_h	±0.25m	1.7m/s	25m/s^2
z_h	±0.22m	1.6m/s	35m/s^2
Roll, ψ_h	±20deg	135deg/s	2500deg/s^2
Pitch, θ_h	±20deg	130deg/s	2000deg/s^2
Yaw, ϕ_h	±20deg	135deg/s	3000deg/s^2

3. Non-linear MPC for Motion Cueing

The proposed Motion Cueing Algorithm is based on Model Predictive Control (MPC), and it is the development of the one described in [Bru2, Mar1]. The advantages of MPC paradigm are well known: the procedure solves at each step a constrained, optimal control problem over a prediction window, applies the first element of the computed solution and iterates again, so that the effect of uncertainties in the model and of disturbances can be countereacted. Availability of a satisfactory model of the system under control plays a fundamental role in this approach. The effectiveness of the model is strictly related to the presence of constraints. In fact, limitations on the different parameters of the system can be imposed, that are taken into account when solving the minimization step. In this way, the system behaviour can be defined by acting on quantities that have a physical meaning, leading to a more practical setup with respect to more traditional approaches. Analogously, the presence of a weighted cost function [Wan1] defines the tuning procedure, making possible the regulation of performance by acting on the weights themselves.

From the implementation point of view, the optimization procedure is the core of the MPC procedure, in particular when dealing with non-linear MPC (NMPC). In the specific framework, there are strict real-time requirements, involving fast dynamics (the requested control frequency is 100 Hz), hence fast computation is crucial. From these considerations, for our application the choice has to be done among fast NMPC tools.

3.1. Model

One of the distinctive elements of adopted approach is the modelling of the human perceptive system, i.e. the dynamics of the set of organs that are responsible for the perception of linear acceleration and angular velocities [Bru1, Bru2, Mar1]. Each perceptive degree of freedom is represented by a linear, continuous-time, second order system, derived from the aerospace literature [Hou1] with parameters adapted to the automotive contest. The state-space representations of each organ are then combined to get the complete otoliths and semicircular channels systems, named Σ_O and Σ_S, respectively. The former takes as input the vehicle linear accelerations to produce the perceived ones, the latter acts in the same way with the rotational velocities.

The otoliths system Σ_O must then be modified to introduce the *tilt-coordination* effect. The low-frequency components of accelerations cannot be reproduced within the limited space of a simulation platform, by only using linear displacements: the state-of-the-art workaround is known as tilt-coordination, according to which, the gravitational force is used to reproduce the low frequency components of accelerations, by appropriately tilting the device. To achieve a correct perception the driver's frame rotation has also to be taken into consideration. When using large yaw values, that frame is rotated by a non-negligible angular displacement with respect to the inertial one, hence the inertial frame accelerations would be incorrectly reproduced on the driver if they are not projected in the correct way. Let a be the acceleration the driver is subject to, and a_x, a_y, a_z its components along the inertial reference system. If θ and ψ are the pitch and roll angles, respectively, we have that the gravity vector g_{TILT} of the non-inertial system moving together with the eye-point of the driver is obtained by rotating the inertial gravity vector as

$$g_{TILT} = R(\theta) \cdot R(\psi) \cdot \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} -g \sin\theta \\ g \cos\theta \sin\psi \\ g \cos\theta \cos\psi \end{bmatrix} \quad (1)$$

On the other hand the driver's reference frame transformation can be characterized by

$$a_d = \begin{bmatrix} a_x \cos\phi - a_y \sin\phi \\ a_x \sin\phi + a_y \cos\phi \\ a_z \end{bmatrix} \quad (2)$$

where a_d is the acceleration on the driver's frame. By combining a_d with Eq. 1, we obtain that with the use of tilt coordination, if suffices to provide the driver with the specific acceleration $\tilde{a} = a_d - g_{TILT}$,

$$\tilde{a} = \begin{bmatrix} a_x \cos\phi - a_y \sin\phi + g \sin(\theta) \\ a_x \sin\phi + a_y \cos\phi - g \cos\theta \sin\psi \\ a_z - g \cos\theta \cos\psi \end{bmatrix} \approx \begin{bmatrix} a_x + g\,\theta \\ a_y - g\psi \\ a_z - g \end{bmatrix} \quad (3)$$

where a linear approximation is used. The small-angles approximation in Eq. 3 is acceptable in the situation at hand. Tilt coordination provides an essential contribution to the effectiveness of a dynamical platform, and must be well managed. In this sense, it has to be combined with visual and audio clues to foul the driver perceptive system, and, at the same time, the simulator has to rotate slowly enough not to trigger the semicircular channels reaction. This is a constraint that has to be taken care of in the problem formulation. Following the approach in [Bru2, Mar1], this is achieved by augmenting the state of the otolithic system to include the effect of the platform inclinations, obtaining the "augmented" systems $\Sigma_{\bar{O}}$.

Σ_S and $\Sigma_{\bar{O}}$ are then combined to get the complete vestibular model. Model inputs are the actual vehicle accelerations and rotational velocities, and its outputs are the corresponding perceived quantities. The motion control systems of most devices require to specify the positions of the eye-point, and not accelerations and rotations, therefore linear velocities v_i and positions p_i, $i = x,y,z$ are obtained by using a simple integral subsystem $\dot{x}_I = Ax_I + Ba$, where

$$x_I = \begin{bmatrix} p_x & v_x & p_y & v_y & p_z & v_z \end{bmatrix}^T, \; a = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T \quad (4)$$

the same is applied to calculate the yaw position ϕ from $\dot{\phi}$.

The complete model has the following state, input and output vectors

$$x_V = \begin{bmatrix} x_S^T & x_{\bar{O}}^T & x_I^T \end{bmatrix}^T \in \mathbb{R}^{21} \quad (5)$$

$$u_V = [a^T \;\; \beta^T]^T \in \mathbb{R}^6 \quad (6)$$

$$y_V = \begin{bmatrix} y_S^T & y_{\bar{O}}^T & \beta^T & x_I^T & \dot{\beta}^T \end{bmatrix}^T \in \mathbb{R}^{18} \quad (7)$$

where $\beta = [\psi \;\; \theta \;\; \phi]^T$.

With respect to the implementation described in [Bru2, Mar1, Mar2], the mechanical structure of DiM introduces an increase in the complexity, due to some non-linear aspects and the increase in the model dimension. As seen in Fig. 1, the platform can be thought as the combination of two dynamical subsystems, the hexapod and the tripod. This specific constructive choice introduces a "redundancy" in three DOFs, i.e. longitudinal, lateral and yaw displacements. Both the tripod and the hexapod contribute to all of those displacements, but in a different way depending on the tuning of the algorithm, as will be explored in the next section. For instance, one could set the parameters so that the low frequency components of the yaw displacement are reproduced by the tripod (which has a larger operational space) while the high frequency ones are managed by the hexapod. Longitudinal, lateral accelerations a_x, a_y, and yaw velocity $\dot{\phi}$ can then decomposed as

$$a_x = a_{x,t} + a_{x,h}, \quad a_y = a_{y,t} + a_{y,h}, \quad \dot{\phi} = \dot{\phi}_t + \dot{\phi}_h \quad (8)$$

were indexes t and h denote tripod and hexapod components, respectively. The input vector is now of size 9. As a consequence, the integrated velocities and positions in the model are split as well, yielding to 5 additional states (linear positions and velocities, yaw position). The complete system is now of size 26 with 9 inputs, posing quite a challenge from a computational point of view. A further, relevant modification with respect to the linear case is the reformulation of Eq. 3. Given the extended yaw range available with the DiM platform compared to the one described in [Bru2], $\tilde{a}$ must be rewritten as

$$\tilde{a} = \begin{bmatrix} a_x \cos\phi - a_y \sin\phi + g\,\theta \\ a_x \sin\phi + a_y \cos\phi - g\psi \\ a_z - g \end{bmatrix} \quad (9)$$

With this modification the longitudinal and lateral subsystems cannot be considered linear anymore.

Note that the model derived in this way does not take into account any dynamic information about the platform's actuators. This is due to the high complexity of the device. A closed form expression for the hexapod dynamics can be derived only in few, specific situations [Yan1] that do not include the one at hand, that is further complicated by the integration with the tripod. The non-linearities due to the actuators behaviour and their reciprocal interferences are managed through the introduction of specific constraints, derived from the inverse kinematics as specified in the next section.

3.2. Constraints

The constraints the system is subject to, and that have to be considered when solving the optimization problem, are basically two, due to:

- Maximum and minimum length admissible for each of the nine actuators (six on the hexapod, three on the tripod);
- Interference avoidance between hexapod and tripod actuators.

By only providing the limitations on the actuators length to the solver, feasibility of the problem it is not yet guaranteed. Actuator physical dimensions and reciprocal positions also impose the fulfilment of a non-interference constraint to avoid configurations not reachable by the actuators, due to their not negligible size.

This second point is more critical, affecting the integrity of the platform instead of dealing with driver comfort. Due to the necessity of maintaining a feasible real time procedure only this latter constraint is taken into account.

To obtain an analytical solution one should derive a closed form expression of the admissible space. Such a task is a challenging one, therefore it is proposed to use instead an approximated analytical surface.

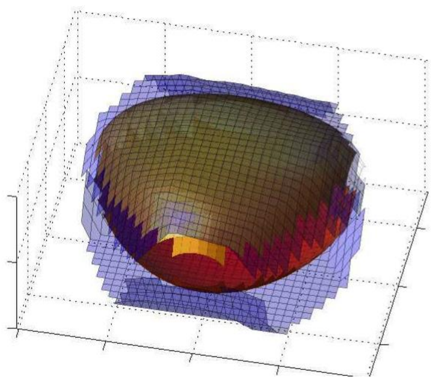

Fig. 2. Closed form surface fitting

First, a (dense) set of measurements is collected by mapping the maximum tripod yaw ϕ_t for each possible couple x_t, y_t (Fig. 2). This $\mathbb{R}^2 \to \mathbb{R}$ map is then approximated with an implicit function of the form

$$ax_t^2 + by_t^2 + cx_t^2 y_t + dx_t y_t^2 + ex_t y_t + f\phi_t + g = 0 \quad (14)$$

and the parameters *a,…,d* computed with a standard curve-fitting procedure. The resulting non-linear function can be directly used as constraint during optimization.

3.3. Optimization

Among the several fast NMPC toolboxes, `ACADO toolkit` [Hou2] has been chosen for the following reasons:

- it is intuitive and easy to use;
- it is open-source;
- it supplies an automatic code generator for fast implementations;
- it deals with mixed linear/non-linear models.

The optimization function is

$$\min_{\substack{x_0,\dots,x_N \\ u_0,\dots,u_{N-1}}} \sum_{k=0}^{N-1} \|h(x_k,u_k) - \tilde{y}_k\|_{W_k}^2 + \|h_N(x_N) - \tilde{y}_N\|_{W_N}^2 \quad (15)$$

$$\begin{aligned}
\text{s.t.}\quad & x_0 = \hat{x}_0 \\
& x_{k+1} = F(x_k, u_k, z_k), && \text{for } k = 0,\dots,N-1 \\
& x_k^{lo} \le x_k \le x_k^{up}, && \text{for } k = 0,\dots,N \\
& u_k^{lo} \le u_k \le u_k^{up}, && \text{for } k = 0,\dots,N-1 \\
& r_k^{lo} \le r_k(x_k, u_k) \le r_k^{up}, && \text{for } k = 0,\dots,N-1 \\
& r_N^{lo} \le r_N(x_N) \le r_N^{up}
\end{aligned}$$

where $x \in \mathbb{R}^{26}$ denotes the differential state, $u \in \mathbb{R}^9$ the control input, $z \in \mathbb{R}^9$ the algebraic variables, and value 0 for the time index k denotes the current time. h and h_n are the reference functions and $W_k, W_N \in \mathbb{R}^{35}$ are the weighting matrices. $\tilde{y}_k$ and $\tilde{y}_N$ denote the time varying reference. $(\cdot)^{lo}$ and $(\cdot)^{up}$ denote the lower and upper bound of the relative variable and r_k, r_N are the constraint function applied along the horizon window and on the final term, respectively. Finally, F defines an ordinary differential equation.

A useful feature of `ACADO` is that it allows mixing linear and non-linear models exploiting the linearity to improve performance [Qui1]. A convenient reformulation of the problem requires writing the model in two parts:

- Vertical and yaw DOFs compose a linear submodel
- Longitudinal and lateral DOFs has a (partial) non-linear description.

As can be seen from Eq. 15, reference trajectories have to be provided to 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, differently from the classical washout approach, constant zero references for the position of the all the six DOFs and for the velocities of the longitudinal ones are used. 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.

The integration of gravity effect into the model as described in Subsection 3.1 automatically introduces tilt coordination as an effect of the optimal control procedure. Performance can also be increased by using long prediction/control windows. However, this is hard to accomplish due to the difficulty in getting reliable information on the future driver's behaviour, and to the hard real-time computational constraints. Hence, in this first implementation a short, constant-valued prediction window is used.

Compared to linear case described in [Bru2, Mar1] the use of non-linear MPC introduces other advantages, such as:

- When using large values of yaw angle, it is possible to precisely take into account the lateral/longitudinal component needed to correctly reproduce acceleration on the driver's frame, while fulfilling the constraints;
- It is straightforward to combine the tripod and hexapod using weight of the cost function;
- Avoiding interference can be obtained by simply imposing a proper constraint;

4. Results

In this Section, some simulation results are presented. The simulated vehicle is a GT class car, and the virtual test track is a digital version of the Calabogie track. In order to better explain the advantages of the proposed procedure the longitudinal, lateral and yaw interaction during big tripod movements is reported. Also, due to space constraints, values of the tuning parameters are omitted. The MCA is set-up so that platform working area is exploited at best.

Fig. 5 illustrates the value of the non-interference constraint (Eq. 14) during the simulation. After 10 and 20 seconds its limit is reached, and consequently the yaw action is transferred from the tripod to the hexapod (Fig. 4), in order to keep the platform within its operational space. The cost of this manoeuver is a negligible decrease of the perceived yaw velocity tracking performance (Fig. 3), which is almost imperceptible to the driver. The advantage in terms of device exploitation, safety management and motion reliability is clear.

From the computational point of view, the simulations have been performed on a standard Intel i5 2.4 GHz, OSX 10.9, with an average computation time of about 1x the required one (control frequency of 100 Hz). Further study is required to test the proposed solution with dedicated hardware in order to guarantee real time in any possible situation.

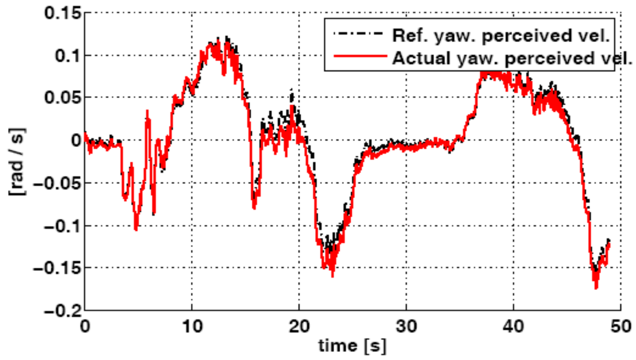

Fig. 3. Perceived Yaw velocity tracking

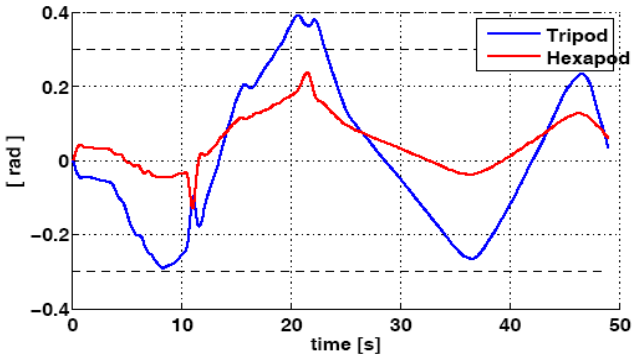

Fig. 4. Yaw displacement

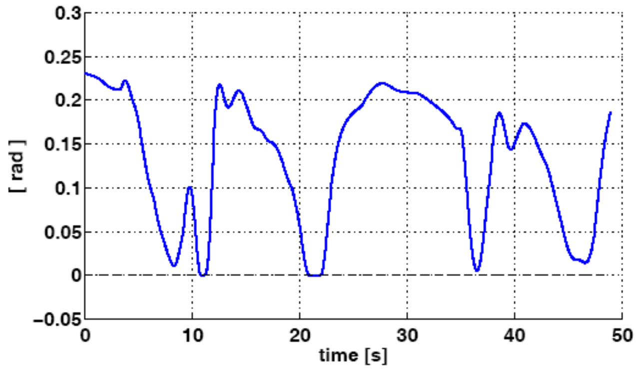

Fig. 5. Non-interference constraint value

5. Conclusions

In this paper a MC algorithm for a high performance, 9 DOFs dynamic simulator has been described, which is based on non-linear MPC techniques. The algorithm represents an important improvement w.r.t. the linear algorithm described in [Bru2], and it allows handling the complex platform mechanical structure in a natural way. The present algorithm is based on a perception model, and it exploits a partial correct inverse kinematic characterization of the platform. Simulation results show that satisfactory performance can be achieved in terms of reproducing accurate perception even if in particularly critical operating conditions.

6. References

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

[Aug1] B. Augusto, R. Loureiro. Motion cueing in the Chalmers driving simulator: A model predictive control approach. Master's thesis, Chalmers University of Technology, 2009.

[Dag1] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, N. Ma zi. Model-based predictive motion cueing strategy for vehicle driving simulators. Control Engineering Practice, vol. 17, no. 9, pp. 995–1003, 2009.

[Bru1] M. Bruschetta, F. Maran, M. Baseggio, A. Beghi, D. Minen. An mpc approach to the design of motion cueing algorithms for driving simulators. Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, pp. 692 –697.

[Bru2] M. Bruschetta, F. Maran, A. Beghi. A real time implementation of mpc based motion cueing strategy for driving simulators. Decision and Control (CDC), 2012 51st International IEEE Conference on, dec. 2012.

[Mar1] F. Maran. Model based control techniques for automotive applications. Ph.D. dissertation, University of Padova, February 2013.

[Mar2] F. Maran, M. Bruschetta, A. Beghi. A real-time implementation of an mpc-based motion cueing strategy with time-varying prediction. Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on. IEEE, 2013, pp. 4149–4154.

[Gar1] N. J. Garrett, M. C. Best. Model predictive driving simulator motion cueing algorithm with actuator-based constraints. Vehicle System Dynamics, vol. 51, no. 8, pp. 1151–1172, 2013.

[Fre1] R. Frezza, A. Saccon, D. Bacchet. Smartdriver: a sensor based model of a car driver for virtual product development. Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on, vol. 1, july 2003, pp. 366 – 370.

[Hou1] J. A. Houck, R. J. Telban, F. M. Cardullo. Motion cueing algorithm development: Human-centered linear and nonlinear approaches. NASACR, vol. 213747, May 2005.

[Yan1] J. Yang, Z. Geng. Closed form forward kinematics solution to a class of hexapod robots. Robotics and Automation, IEEE Transactions on, vol. 14, no. 3, pp. 503–508, Jun 1998.

[Hou2] B. Houska, H. J. Ferreau, M. Diehl. Acado toolkitan open-source framework for automatic control and dynamic optimization. Optimal Control Applications and Methods, vol. 32, no. 3, pp. 298–312, 2011.

[Qui1] R. Quirynen, S. Gros, M. Diehl. Efficient nmpc for nonlinear models with linear subsystems. Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, Dec 2013, pp. 5101–5106.