



**HAL**  
open science

# MPC delay compensation based on maximal controllable sets for real-time driving simulators

Martin Soyer, Sorin Olaru, Zhou Fang

## ► To cite this version:

Martin Soyer, Sorin Olaru, Zhou Fang. MPC delay compensation based on maximal controllable sets for real-time driving simulators. 25th International Conference on System Theory, Control and Computing (ICSTCC 2021), Oct 2021, Iași, Romania. 10.1109/ICSTCC52150.2021.9607083. hal-03408219

**HAL Id: hal-03408219**

**<https://hal.science/hal-03408219>**

Submitted on 28 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# MPC delay compensation based on maximal controllable sets for real-time driving simulators

Martin Soyer

Laboratoire des signaux et systèmes  
Université Paris-Saclay, CentraleSupélec  
Gif-sur-Yvette, France  
martin.soyer@centralesupelec.fr

Sorin Olaru

Laboratoire des signaux et systèmes  
Université Paris-Saclay, CentraleSupélec  
Gif-sur-Yvette, France  
sorin.olaru@centralesupelec.fr

Zhou Fang

AD Simulation & VR center  
Renault Technocentre  
Guyancourt, France  
zhou.fang@renault.com

**Abstract**—This paper addresses the constrained control in relationship with time-delayed systems application. We are particularly interested in its potential application to driving simulation. Driving simulators are electro-mechanical structures that reproduce the environment of a driver (steering wheel, pedals, seat, road displayed on a screen,...) and are able to move within a restricted safe area on purpose to reproduce the driver’s feelings and particularly the acceleration ones through the inner ear. Those movements are controlled by the Motion Cueing Algorithm (MCA) which consists in a Model-based Predictive Controller (MPC) converting in closed loop the expected acceleration to displacements all by observing the physical constraints. Its design is realized in a way to minimize the difference between effective felt acceleration and the expected one. In this application context, the movements inertia and communication protocols may induce delays between the computation block of the controller and the response that lead to serious discomforts of the humans in the simulators, known under the *Motion Sickness* symptoms. In the present paper, we revisit the control formulation for this motion cueing algorithm in a control perspective and propose a control strategy that compensates the delays in way to satisfy real-time constraints during implementation in a simple yet effective manner. The performance and limitation of a such strategy are discussed to raise the awareness on the design trade-off and decisions.

**Index Terms**—Predictive Control, Control Systems Design, Real Time Applications

## I. INTRODUCTION

Real-time control of industrial process practically implies delays between signal generation and response measurement with well known implication in the complexity and stability of the underlying processes [1]. Those delays are of different natures: actuators inertia, communication or command computation in the optimal control case. They can be disruptive, particularly for critical real-time systems where signals have to be sent at the right moment.

The present paper is devoted to control of high performance driving simulators [2] which purpose is to reproduce as close as possible the driver feelings by the platform movements in a restrictive and safe workspace (the range of the maneuvering area is around  $10\text{m}^2$ ). Car manufacturers have first focused on visual stimuli with display of virtual road environments on a screen in front of the driver, then they have add motion actuators to stimulate equilibrium feelings sensed in the inner ear. The displacements are deduced from the expected accel-

eration, computed by a software implementing the real vehicle dynamics, by a function called Motion Cueing Algorithm (MCA) [3].

In this framework, acceleration restitution induces discomfort when not synchronized with visual stimuli. Those drawbacks are know in the literature as Motion Sickness, who received the attention of different communities in the last 50 years, with the common goal of translating the underlying phenomena towards mathematical models [4]–[6]. The reason for such an interest in the mathematical modeling of the motion sickness is simple: such a model will enable its predictions and ultimately lead to strategies for its mitigation.

While driving simulators controllers were initially designed with open-loop filter-based structures, most of the publications nowadays focus on closed loop algorithms based on real-time management of the prediction such as Model Predictive Control (MPC) [7]–[10] in order to enhance the constraints management and avoid some sickness features such as backlash effects [11]. The real-time control of such platform which comes with important safety constrained leads to computationally involving schemes. Some improvements have been proposed in term of complexity reduction [12] and recently explore alternative schemes such as neural network solver implementation [13]. However, aside the constraint handling and the computational limitations, the inherent delays are particularly difficult to handle in the control design with direct implications in the motion sickness phenomena. The goal of this work is to propose a control design which prevent the driver from feeling unease, by dealing with the time-delay from the design stage in the predictive control framework.

The classical approach for the delay compensation consists in translating the problem in an extended state space representation using past control actions and then design a MPC controller for the undelayed resulting model [14]. However, this approach is facing computational limitations particularly for a real-time implementation due to the curse of dimensionality which emerges with the extended state space and become more difficult to overcome when the delay increases. Most publications deal with this issue by proposing set-invariance based perspective [14], [15] or investigate an approach close to the Smith predictor philosophy [16].

In this work, we propose an MPC design based on maximal

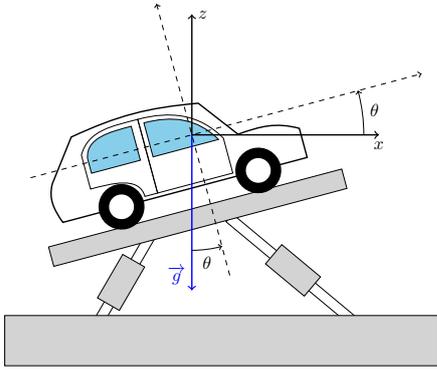


Fig. 1. Driving simulator scheme

controllable set for delayed linear systems in the same way of MPC alleviation philosophy for undelayed ones. The background of driving simulators modelization and delay compensation are addressed in Section II. The Maximal controllable set approach is developed in section III. The framework and simulation results are given in Section IV and discussed in Section V.

## II. PROBLEM FORMULATION

The schematic view of the dynamic platform is depicted in the Figure 1. The basis of the structure (inclined gray plate in the figure) can move longitudinally and laterally by means of the integrated rails and it can be tilted by electrical actuators which equip the hexapod. Those different physical actuators are controlled with a low level (internal) feedback law which tracks position (for the rails) and tilt (for the hexapod) references. These reference signals are provided by the Motion Cueing Algorithm (MCA) which is the block that concerns the effective design of this work.

The dynamics of a high performance<sup>1</sup> driving simulator can be summarized with the following time-delay discrete-time linear model obtained by the discretization with the sampling time  $T_s$  (considered in the present application for the available benchmark at 10 ms):

$$x(k+1) = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ T_s & 0 \\ 0 & T_s \end{bmatrix} u(k-d) \quad (1)$$

$$y(k) = [0 \quad 0 \quad g] x(k) + [1 \quad 0] u(k-d)$$

The state vector  $x(k) = [p(k) \quad v(k) \quad \theta(k)]^T$  includes three physical components: position, speed and tilt angle while the output  $u(k) = [u_l(k) \quad u_r(k)]^T$  corresponds to the linear lateral/longitudinal<sup>2</sup> acceleration of the rails and the tilt rate,  $g$  denotes the gravitational acceleration.

<sup>1</sup>The *high-performance* is understood here in the sense that the low level control of the longitudinal/lateral/tilt movements is tuned to deliver a fast and accurate responses.

<sup>2</sup>We include here for brevity one axis state space model (e.g. lateral), knowing that a counterpart (1) exists for the other axis (e.g. longitudinal).

Practically, the simulator integrates under the workspace constraints the acceleration information provided by the software module which generates the virtual scene. Here we assume the workspace constraints to be symmetrical with respect to their origin:

$$|p(k)| \leq p_{max}, \quad (2)$$

$$|v(k)| \leq v_{max}, \quad (3)$$

$$|\theta(k)| \leq \theta_{max}. \quad (4)$$

The power supply and the maximal tilt rate felt by a person restricts the inputs equally:

$$|u_l(k)| \leq u_{l,max}, \quad (5)$$

$$|u_r(k)| \leq u_{r,max}. \quad (6)$$

Thus, as the considered constraints are polyhedral, in the following we will consider the compact formulation  $\mathbb{X} = \{x \in \mathbb{R}^n \mid F_x x \leq g_x\}$  for the state constraints and  $\mathbb{U} = \{u \in \mathbb{R}^m \mid F_u u \leq g_u\}$  for input constraints.

**Remark.** The model (1) is obtained from the linearization of the nonlinear dynamics whose response is  $y(k) = g \sin(\theta(k)) + u_l \cos(\theta(k))$  and corresponds to the acceleration felt by the driver in his/her frame. The assumption of small angles is valid in automotive driving simulators which have limited variations with respect to other fields of application involving immersive virtual reality.

Time-delays are issued by communication protocol and the time-response of the low-level platform dynamics [17] and can be taken into account in the model, with a time-invariant value  $d \in \mathbb{N}$ .

In this paper, we assume the both actuation channels are associated to the same delay. This can always be enforced by considering  $d$  to be the upper bound of delays on the respective actuation channels. In practice, even if they present slightly differences, the delays are in the range (50-100 ms).

The Figure 2 depicts the operating principles for the control system (1). The objective is to provide an accurate tracking of the acceleration signal.

**Remark.** The acceleration signal (output of the system) is in direct relation with the input signal and thus the control loop can be formalized as a tracking problem for an (over-actuated) system. This trivial problem meets however the physical limitations of the state constraints. As such, most acceleration profiles need to be replaced by admissible trajectories all by minimizing the Motion Sickness (which is related principally to the time-lag but also to the tracking error).

**Remark.** The acceleration trajectory to be tracked is available at the current moment in time and its extrapolation is subject to important uncertainties due to poor predictability of human reactions in relationship with the scene and the motion cueing algorithm. In control related notions, this means that the future trajectory  $\{a_{k+1|k}, a_{k+2|k}, a_{k+3|k}, \dots\}$  can be made available but it cannot be granted to be receded at the next time instant (i.e.  $a_{k+j|k+i}$  may consistently differ from

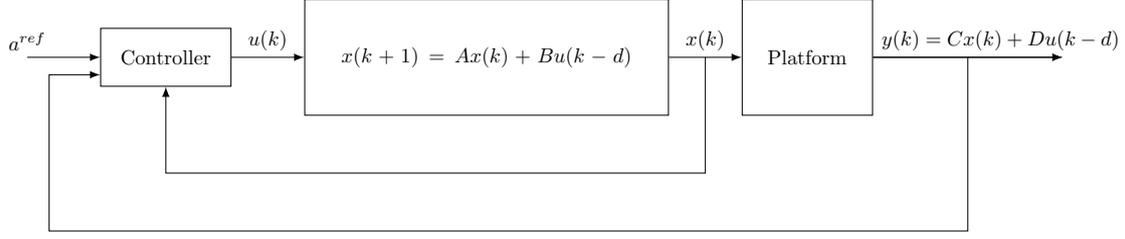


Fig. 2. Block diagram of the control structure operating

$a_{k+j|k+i+1}$  for  $j > i + 1 > 0$ ). The accurate tracking of a particular sequence can turn to a poor performance in case of trajectory update from the MCA. The first conclusion in this respect is that control design has to ensure a recursive feasibility mechanism independent of the assumptions on the tails of the future trajectories.

Taking into account also the time-delay particularity of the dynamical model, this last remark impose recursive feasibility requirements exclusively based on the current state and the previous control inputs.

The classical approach for the constrained control design for time-delay systems is the predictive control (MPC) using an extended state-space model. This state-space prediction model includes the past control actions in the extended state vector

$$\xi(k) = [x(k) \quad u(k-d) \quad \dots \quad u(k-1)]^T.$$

With this artefact, the extended dynamics become:

$$\begin{aligned} \xi(k+1) &= A_\xi \xi(k) + B_\xi u(k) \\ y(k) &= C_\xi \xi(k) \end{aligned} \quad (7)$$

where:

$$A_\xi = \begin{bmatrix} A & B & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{I} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & & \ddots & & \vdots \\ \mathbf{0} & \dots & \dots & \dots & \mathbb{I} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{0} \end{bmatrix}, B_\xi = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{I} \end{bmatrix} \quad (8)$$

$$C_\xi = [C \quad D \quad \mathbf{0} \quad \dots \quad \mathbf{0}]$$

Then classical MPC formulation of the tracking problem is:

$$\begin{aligned} \min_{\substack{(u(k), \dots, \\ u(k+N-1))}} & \sum_{i=1}^N \|y^{ref}(k+i) - y(k+i)\|_{q_y}^2 \\ & + \|\xi(k+i)\|_{Q_x}^2 + \|u(k+i)\|_R^2 \\ \text{s.t.} & \xi(k+1) = A_\xi \xi(k) + B_\xi u(k) \\ & y(k) = C_\xi \xi(k) \\ & (\xi(k+1), \dots, \xi(k+N-1)) \in \mathbb{X} \times \mathbb{U}^d \\ & (u(k), \dots, u(k+N-1)) \in \mathbb{U} \\ & \xi(k+N) \in \Omega_f \end{aligned} \quad (9)$$

where  $q_y$ ,  $Q_x$  and  $R$  are weighting matrices,  $N$  is the prediction horizon,  $y^{ref}$  is the reference signal and  $\Omega_f$  is the

terminal positively invariant set and needs to be parameterized according to an admissible trajectory. We recall the definition of this property below for the class of linear discrete-time systems with linear feedback structure which is of interest in the current framework.

**Definition II.1.** A set  $\Omega$  is positively invariant with respect to the delay-free dynamics  $x(k+1) = Ax(k) + Bu(k)$  subject to a linear feedback law  $u(k) = -Kx(k)$  if for any initial state  $x(k) \in \Omega$  it follows  $(A - BK)x(k) \in \Omega$  and the control is admissible  $-Kx(k) \in \mathbb{U}$ .

For the linear case, this set can be constructed with the procedure given in [18]. Unfortunately, the formulation (9) is too complex for the real-time application considered here, the larger extended state space size running against the curse of dimensionality on one hand, and on the other hand the parametrization of the terminal set being an additional burden on the computational side.

In the following, we propose a strategy based on the direct use of the Maximal Controllable Set (MCS) or its approximation. This set defined as the largest set of initial states for which there exists an admissible sequence of control actions that makes the state trajectory to remaining in the set itself. In the following, this set will be denoted  $\mathcal{C}$ . In practice, this set can be approximated by the  $N$ -step Controllable Set  $\mathcal{C}_N$  in polyhedral form. It can be construct from a positively invariant set through the procedure in [19].

### III. MAXIMAL CONTROLLABLE SET BASED APPROACH

In this section, we adress the alleged control structure for time-delay linear systems applied to the driving simulation application. This MPC-based structure can be declined in two versions, one using the memory of the past inputs, the other using the last free state.

For the theoretical analysis of the control scheme, we consider the following generic LTI time-delay system with single output:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k-d) \\ y(k) &= Cx(k) + Du(k-d) \end{aligned} \quad (10)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{1 \times n}$  and  $D \in \mathbb{R}^{1 \times m}$ .

The principle of this method is summarized in Figure 3:

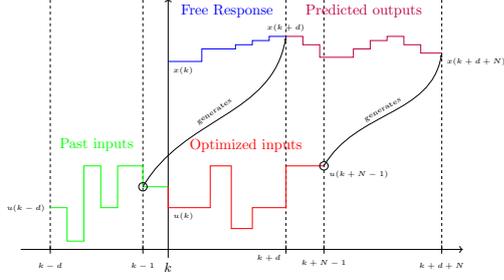


Fig. 3. Principle of the strategy proposed in Section III, the past inputs impose the free response up to  $k+d$  while the optimized inputs are chosen such that the predicted inputs ensure the first predicted state (at stage  $k+d+1$ ) to belong to the controllable set  $\mathcal{C}$

- knowing the past inputs construct the free-response up to  $x(k+d|k)$
- minimize the difference between the reference and the predicted output over the horizon  $k+d, \dots, k+N$
- impose as hard constraint  $x(k+d+1|k) \in \mathcal{C}$ .

The receding horizon optimization related to the MPC formulation is given below:

$$\begin{aligned}
 \min_{(u(k), \dots, u(k+N-1))} & \sum_{i=d+1}^{d+N} \|y^{ref}(k+i) - y(k+i)\|_{q_y}^2 \\
 & + \|x(k+i)\|_{Q_x}^2 + \sum_{i=1}^N \|u(k+i)\|_R^2 \\
 \text{s.t.} & \\
 x(k+i+1) &= Ax(k+i) + Bu(k+i-d) \\
 & \quad \forall i \in \{0, \dots, d+N-1\} \\
 y(k+i) &= Cx(k+i) + Du(k+i-d) \\
 & \quad \forall i \in \{d+1, \dots, d+N\} \\
 x(k+d+1) &\in \mathcal{C} \\
 (u(k), \dots, u(k+N-1)) &\in \mathbb{U}
 \end{aligned} \tag{11}$$

where  $q_y$ ,  $Q_x$  and  $R$  are weighting matrices.

The first component of the solution is effectively applied. Those constraints guarantee the recursive feasibility of the controller while optimizing the cost function. The recursive feasibility of the controller can be guaranteed using the following results.

**Lemma III.1.** *Let  $\mathcal{C}$  be an admissible controlled invariant set with respect to the delayed-free dynamics:  $\xi(k+1) = A\xi(k) + Bv(k)$ , where  $(A, B)$  are the matrices of (10). If the current state  $x(k) \in \mathcal{C}$  and the past inputs of (10)  $\{u(k-d), \dots, u(k-1)\} \in \mathbb{U}$  such that  $\{x(k+1), \dots, x(k+d)\} \in \mathcal{C}$ , then there exists a control action  $u(k)$  such that  $x(k+d+1) = Ax(k+d) + Bu(k) \in \mathcal{C}$ .*

*Proof.* Let us start from the assumption that  $x(k) \in \mathcal{C}$ . By exploiting the existence of feasible control sequence such that

$\{x(k+1), \dots, x(k+d)\} \in \mathcal{C}$  one can concentrate on the delay-free dynamics:

$$\xi(k+1) = A\xi(k) + Bv(k)$$

which is equivalent to due to time-invariance to

$$x(k+d+1) = Ax(k+d) + Bu(k)$$

For any  $\xi(k) \in \mathcal{C}$  there exists  $v(k) \in \mathbb{U}$  such that  $\xi(k+1) \in \mathcal{C}$ . By choosing  $u(k) = v(k)$  one has  $x(k+d+1) \in \mathcal{C}$ .  $\square$

**Proposition III.1.** *Given an initial state  $x(0) \in \mathcal{C}$  such that  $x(0) = [p_0 \ 0 \ \theta_0]^T$  with  $(p_0, \theta_0) \in \mathbb{R}^2$  and a past control sequence  $\{u(-d), \dots, u(-1)\} = \{0, \dots, 0\}$ , then the algorithm is recursively feasible.*

*Proof.* Since no actions are provided to the plant, the trajectory remain at the origin during  $d$  time steps. Since the origin belongs to the maximal controllable set, the previous lemma can be applied.  $\square$

#### IV. SIMULATION

The simulated driving scenario investigate here is a slalom phase followed by a turn. Here, we only observe the lateral accelerations and movements as well as the MCA responses. The weight parameters use for the simulation are:  $q_y = 10$ ,  $Q_x = \text{diag}([50 \ 1 \ 1])$ ,  $R = \text{diag}([10 \ 1])$  and the prediction horizon:  $N = 100$ . The simulation was computed with MATLAB 2019a with a processor Intel(R) Core(TM) i7-7700K CPU@4.20GHz 16Go RAM.

The Fig.4 depicts the displacement of the lateral rail, its speed and the tilt angle profile as well as their physical limitations that are actually satisfied along the simulation. The acceleration rendering is compared to the reference profile on Fig.5 where we observe the acceleration globally follows the profile with a lower amplitude by noticing that a part of acceleration is proportional to the tilt angle. An important feature of the acceleration rendering is the restitution of the shape of the reference profile, particularly during the slalom phase (there is no saturation due to inputs constraints), then the driver feels the correct variations. More the delay has been compensated which can cancel the motion sickness.

The acceleration of the rail (represented on Fig.6) is added to the tilt angle contribution. We can observe that the rail contributes more quick components of the slalom phase while the tilt impacts more the slow turn phase (which can be explained by the constraints on tilt rate depicted on Fig.7). Limitations of the rail are underlines on Fig.4 and on Fig.8 where the rail part of the state space is represented (position and speed) with the trajectory within Maximal Controllable Set, which means the dynamics is conservative. Fig.9 depicts the computation time at each iteration.

#### V. DISCUSSION

First, this strategy is limited by the complexity of the Controllable set. Indeed, the complexity of the optimization problem (11) depends on the complexity of  $\mathcal{C}$ . The curse of dimensionality can be expected as the dimension of the

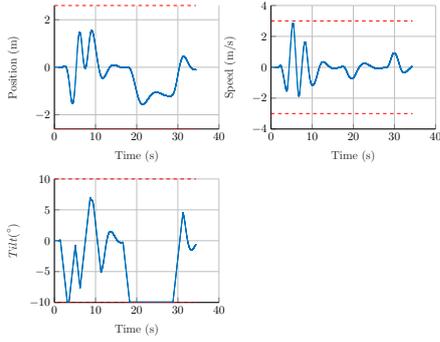


Fig. 4. States trajectory in the time domain: rail position (upper left), rail velocity (upper right), tilt angle (below left)

state-space increases and thus the whole complexity of the procedure making the real-time performance unreachable.

In our application the belonging of the  $(d + 1)th$  state to  $\mathcal{C}$  may imply that the controller forces punctually the trajectory to remain in  $\mathcal{C}$  by applying the maximal input value. This is illustrated in the following variation of the simulation presented in the previous section. Here the design is less conservative, the position component of the cost function has a lower weight:  $Qx = \text{diag}([30 \ 1 \ 1])$ . We can check on the Fig.10,12 that the trajectory uses a larger area and take benefits of the whole workspace to improve the acceleration rendering depicted on Fig.11 with the rail acceleration input and the reference. However, after 20s we can observe an artefact in the rail acceleration input that impacts the response. If this artefact is short enough, it may be filtered by the inner ear and so not impact the driver, but it the sensibility depends on the driver physiology. So there is a trade-off between the conservativeness of the design (weights on states and inputs) and the implication of delayed states constraints.

## VI. CONCLUSION

Time delays are inherent to the structure of driving simulators due to the mechanical inertia and the communication protocol between the algorithms which render the virtual scene and the physical move of the platform, all of them having as a joint action on the human senses. The Motion Sickness being directly related to the time-delay, its inclusion in the control algorithm needs to be optimized in order not to further deteriorate the human perception.

The classical optimal control approach based on extended state space (9) is difficult to implement on real-time systems because of the complexity inherited from the increased dimension of parameters. In this paper we proposed a alleged control strategy based on the knowledge of the maximal controllable set (11) by enforcing the delayed states to remain within this safe set at a reduced computational cost.

This strategy is recursively feasible and globally follows the expected acceleration while satisfying states and inputs constraints. However its performance can be limited on one

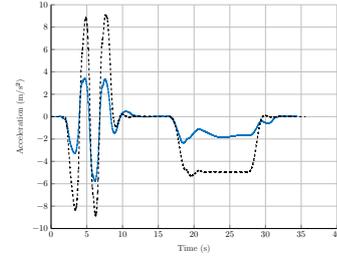


Fig. 5. Acceleration reference profile (dashed), output acceleration (blue)

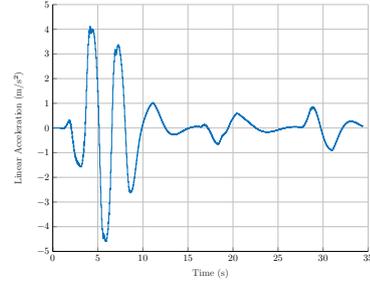


Fig. 6. Rail acceleration input

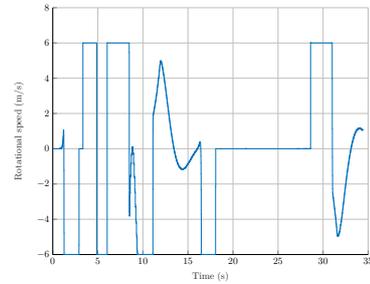


Fig. 7. Tilt rate input

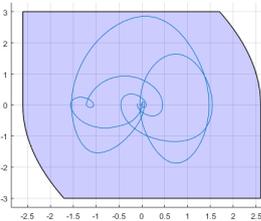


Fig. 8. Trajectory in the state space of the rail components (position, speed), the blue set corresponds to projection of the maximal controllable set on the rail components

side by the size (topology) of controllable set and on the other side by the policy of constraints activation among the control channels. This latter drawback can be overcome with the choice of weights in the design of the strategy.

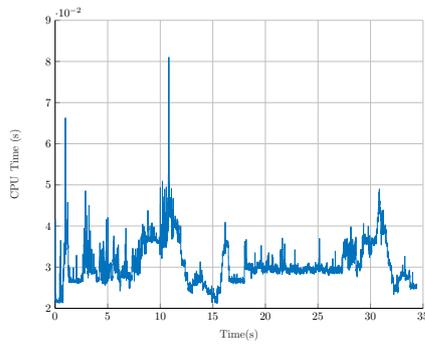


Fig. 9. CPU time of the input computation at each iteration

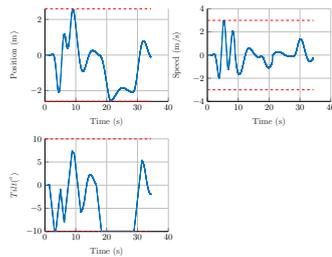


Fig. 10. states trajectory in the time domain: rail position (upper left), rail velocity (upper right), tilt angle (below left)

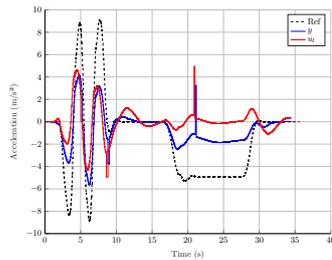


Fig. 11. Comparison between linear rail acceleration (red) and output response (blue)

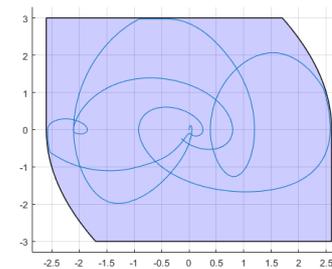


Fig. 12. Trajectory in the state space (rail components)

## REFERENCES

- [1] Silviu-Iulian Niculescu. *Delay effects on stability: a robust control approach*, volume 269. Springer Science & Business Media, 2001.
- [2] Oliver Carsten and A Hamish Jamson. Driving simulators as research tools in traffic psychology. In *Handbook of traffic psychology*, pages 87–96. Elsevier, 2011.
- [3] Jacob A Houck, Robert J Telban, and Frank M Cardullo. Motion cueing algorithm development: Human-centered linear and nonlinear approaches. 2005.
- [4] Michael E McCauley, Jackson W Royal, C Dennis Wylie, James F O’Hanlon, and Robert R Mackie. Motion sickness incidence: Exploratory studies of habituation, pitch and roll, and the refinement of a mathematical model. Technical report, Canyon Research Group Inc Goleta Ca Human Factors Research Div, 1976.
- [5] Takahiro Wada, Satoru Fujisawa, and Shunichi Doi. Analysis of driver’s head tilt using a mathematical model of motion sickness. *International Journal of Industrial Ergonomics*, 63:89–97, 2018.
- [6] JC Guignard and Michael E McCauley. Motion sickness incidence induced by complex periodic waveforms. *Aviation, space, and environmental medicine*, 53(6):554–563, 1982.
- [7] Z Fang, M Tsushima, E Kitahara, N Machida, D Wautier, and A Kemeny. Motion cueing algorithm for high performance driving simulator using yaw table. *IFAC-PapersOnLine*, 50(1):15965–15970, 2017.
- [8] A. Beghi, M. Bruschetta, and F. Maran. A real time implementation of mpc based motion cueing strategy for driving simulators. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 6340–6345, 2012.
- [9] M. Bruschetta, F. Maran, and A. Beghi. A nonlinear, mpc-based motion cueing algorithm for a high-performance, nine-dof dynamic simulator platform. *IEEE Transactions on Control Systems Technology*, 25(2):686–694, 2017.
- [10] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi. Multiobjective and interactive genetic algorithms for weight tuning of a model predictive control-based motion cueing algorithm. *IEEE Transactions on Cybernetics*, 49(9):3471–3481, 2019.
- [11] Zhou Fang and Andras Kemeny. Review and prospects of Renault’s MPC based motion cueing algorithm for driving simulator. In *Proceedings of the Driving Simulation Conference Europe, Paris, France*, pages 4–5, 2014.
- [12] Sarmad Munir, Morten Hovd, Zhou Fang, Sorin Olaru, and Andras Kemeny. Complexity reduction in motion cueing algorithm for the ultimate driving simulator. *IFAC-PapersOnLine*, 50(1):10729–10734, 2017.
- [13] Carolina Rengifo, Jean-Rémy Chardonnet, Damien Paillot, Hakim Mohellebi, and Andras Kemeny. Solving the constrained problem in model predictive control based motion cueing algorithm with a neural network approach. 2018.
- [14] Mohammed-Tahar Laraba, Sorin Olaru, and Silviu-Iulian Niculescu. Linear model predictive control and time-delay implications. *IFAC-PapersOnLine*, 50(1):14406–14411, 2017.
- [15] Mohammed-Tahar Laraba, Sorin Olaru, Silviu-Iulian Niculescu, Franco Blanchini, Giulia Giordano, Daniele Casagrande, and Stefano Miani. Guide on set invariance for delay difference equations. *Annual Reviews in Control*, 41:13–23, 2016.
- [16] Tito LM Santos, Julio E Normey-Rico, and Daniel Limón. Explicit input-delay compensation for robustness improvement in mpc. *IFAC Proceedings Volumes*, 43(2):384–389, 2010.
- [17] Zhou Fang, Gilles Reymond, and Andras Kemeny. Performance identification and compensation of simulator motion cueing delays. *Trends in driving simulation design and experiments*, pages 111–120, 2010.
- [18] Elmer G Gilbert and K Tin Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic control*, 36(9):1008–1020, 1991.
- [19] Hoai-Nam Nguyen, Per-Olof Gutman, Sorin Olaru, and Morten Hovd. Implicit improved vertex control for uncertain, time-varying linear discrete-time systems with state and control constraints. *Automatica*, 49(9):2754–2759, 2013.