Interpolation-Based MCA for acceleration rendering

Martin Soyer, Sorin Olaru, Zhou Fang, Didier Wautier, Kemeny Andras

Interpolation-Based MCA for acceleration rendering

M.Soyer[1,2], S.Olaru[2], Z.Fang[1], D.Wautier[1], A.Kemeny[1]

(1) Renault Technocentre, AD Simulation & VR center, Guyancourt, France, e-mail: {martin.soyer, zhou.fang,didier.wautier,andras.kemeny}@ renault.com
(2) Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France, e-mail: {martin.soyer,sorin.olaru}@ centralesupelec.fr

Abstract - *This paper deals with control design of high performance driving simulator Motion Cueing Algorithms (MCA). The highly constraint environment used by manufacturers to simulate driving experience implies a smart exploitation of the workspace and actuators in order to improve the acceleration feelings rendering to the driver. Since the two past decades, optimization-based motion cueing algorithms have been developed in this purpose, particularly, the Model Predictive Control (MPC) framework provides the handling of constraints applied to the system to find an optimal control action. However, the design of MPC-based MCA is a difficult task due to the theoretical and practical requirements such as stability and recursive feasibility guarantees. This contribution is a preliminary study of the application of a novel optimization-based control technique for MCA called Interpolation-Based Control (IBC). Recently developed, IBC showed similar performance as MPC for regulation problems by decreasing the computation time. In this paper, an extension of IBC to the tracking problem is studied for motion cueing.*

Keywords: *Model Predictive Control, Interpolation-Based Control, Motion Cueing Algorithm, Optimization*

1. Introduction

During the past decades, the need of an improved management of the workspace for dynamical simulators led to a more specific interest to control technique based on optimal control theory. The controlled structure that provides a strategy for displacements of simulator's cabin according to rails trajectories and hexapod tilt angles is called Motion Cueing Algorithm (MCA) and its operating principle is given in Figure 1. Constrained optimization provides a framework that overcome problems induced by filter-based algorithms such as the under-exploitation of the workspace or the backlash effect [Fan12, Fan14, Ven16]. Specifically, the Model Predictive Control (MPC) framework, firstly used for MCA in [Dag04], is actually the most studied in the high-performance driving simulation field, it minimizes a tracking acceleration error on a prediction horizon by handling constraints on workspace and actuators limitations in real-time online process. Optimization-based techniques under constraints are designed in order to consider optimization problems as feasible and the technique as stable [Fan16, Ren19]. The parameters tuning problem remains one of the topics under investigation with optimization-based motion cueing algorithms [Fan14]. The computation time to solve optimization problems became a main subject of attention due to real-time delay induced and some work such as [Mun17] studied the complexity of optimization problems in order to reduce it to accelerate explicit algorithms. Many papers take into account the vestibular system to enhance the acceleration rendering at the risk of increasing the complexity. A degree of freedom to reduce computation time is the solver of the QP problem of MPC ([Fan17]). In the following, an adaptation of a less complex optimization technique in a computation point of view is presented.

This paper is a preliminary study about enhancement of MCAs in order to improve the performance of acceleration rendering during the trajectory tracking in optimization-based MCA. Classical optimization-based algorithms generate delays due to computational complexity, particularly when an anticipation is hard to provide for high frequencies reference signals. The aim of this work is to address this issue by decreasing computational complexity by preserving good performance for acceleration tracking. A solution is to consider the optimization-based control technique called Interpolation-Based Control (IBC) developed recently for regulation ([Ngu11, Ngu13, Ngu14]). This technique handles low complexity optimization problems by guaranteeing recursive feasibility and stability. The aim of this paper is to present an adaptation of Interpolation-Based Control for tracking trajectory (IBT) under constraints and then applying it as a basis of a Motion Cueing Algorithm.

This technique needs less parameters to tune and a significative part of operations are done offline such as construction of the maximal controllable set which is essential ingredient to guarantee recursive feasibility of optimization problems and to exploit additional degrees of freedom as possible. The main conceptual step forward is that MCA handles real-time information during the simulation to perform the acceleration tracking by avoiding constraints manipulations on a long prediction horizon.

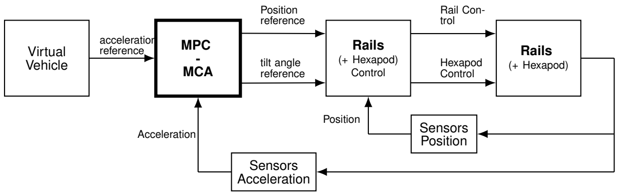

Figure 1: Block diagram of MCA scheme,only bold elements are considered in this paper

2. Problem Formulation

2.1. Models and Performance Criteria

In this preliminary study the classical integrator state-space model will be considerered due to its theoretical significance for the control of rails:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \end{bmatrix} u(k) \\ x(k) &\in \mathbb{X}\ \forall k \\ u(k) &\in \mathbb{U}\ \forall k \end{aligned} \tag{1}$$

where T_s denotes the sampling time. At each sampling time k, $x(k) = [p(k) \quad v(k)]^\intercal$ denotes the vector of position $p(k)$ and speed $v(k)$ while $u(k) = a(k)$ denotes the acceleration of the system according to a direction.$\mathbb{X}$ (resp $\mathbb{U}$) denotes the set of state (resp control) constraints. In the literature, the acceleration is practically considered as a state or an output to be tracked. In this model the acceleration is considered as a control input in order to decrease the dimension of the system and as a main component of the criteria to be optimized. Consider an acceleration reference u_{ref}, then a classical formulation for model predictive controller for MCA with respect to model (1) would be:

$$\begin{aligned} \min_{\substack{u(k),\ldots,\\ u(k+N_h-1)}} \quad & \sum_{k=0}^{N_h} \|u_{ref}(k) - u(k)\|_R^2 + \|x(k)\|_{Q_x}^2 \\ & + \|x(N_h)\|_P^2 \\ \text{s.t.} \quad & x(k) \in \mathbb{X},\ \forall k \in \{1,\ldots,N_h\} \\ & u(k) \in \mathbb{U},\ \forall k \in \{0,\ldots,N_h-1\} \\ & x(N_h) \in \Omega \end{aligned} \tag{2}$$

Where R (resp Q_x) is a weighting matrix on control action (resp on states) , P is the solution of the discrete algebraic Riccati equation relative to the corresponding LQR problem and Ω is the largest positively invariant with respect to the linear feedback :

$$u(k) = -(R + B^\intercal PB)^{-1}B^\intercal PA)x(k) = -Kx(k) \tag{3}$$

The constraint on the last predicted state within Ω guarantees the recursive feasibility of closed loop.

However, in the driving simulation the reference acceleration profile can be a priori unknown and the performance of the MPC controller increases with the prediction horizon N_h, consequently the controller is not practically adapted to small prediction horizons. Moreover the trade-off between prediction length and complexity of the optimization problems implies performing MPC controllers have to handle many constraints and thus impact the optimization solving routine.

Recently, the Interpolation-Based Control (IBC) technique has been developed in [Ngu11, Ngu13] in order to overcome those drawbacks in the regulation framework.

2.2. Interpolation-Based Control

Consider two convex positively invariant sets Ω^o and Ω^v containing the origin and $\Omega^o \subset \Omega^v \subset \mathbb{X}$. At each step k, the measured state $x(k)$ can be decomposed as a convex combination of two states of inner and outer sets $x^o(k) \in \Omega^o$ and $x^v(k) \in \Omega^v$ as it is shown in Figure 2:

$$x(k) = c(k)x^v(k) + (1 - c(k))x^o(k) \tag{4}$$

where $c(k) \in [0, 1]$. The convex factor $c(k)$ and the two components can be computed thanks to the linear programming problem :

$$\begin{aligned} \underset{\substack{(x^v(k),x^o(k),\\ c(k))}}{\text{minimize}} \quad & c(k) \\ \text{subject to} \quad & x^v(k) \in \Omega^v \\ & x^o(k) \in \Omega^o \\ & x(k) = c(k)x^v(k) + (1 - c(k))x^o(k) \\ & c(k) \in [0, 1]. \end{aligned} \tag{5}$$

At optimality, the two components $x^o(k)$ and $x^v(k)$ are located on the frontier of their set as shown on Figure 2 . Thus the control action can be computed as the convex combination :

$$u(k) = c(k)u^v(k) + (1 - c(k))u^o(k) \tag{6}$$

where $u^v(k)$ (resp $u^o(k)$) is the control action of $x^v(k)$ (resp $x^o(k)$)

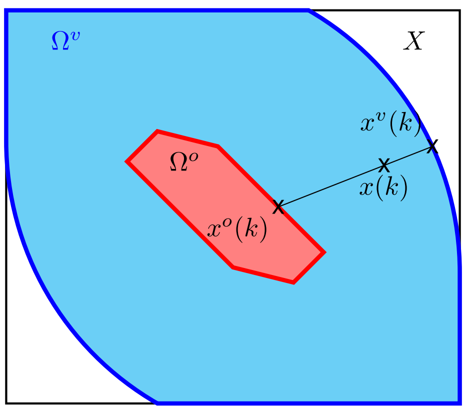

Figure 2: IBC geometric principle

The principle of IBC is summarized in the block diagram of Figure 3.

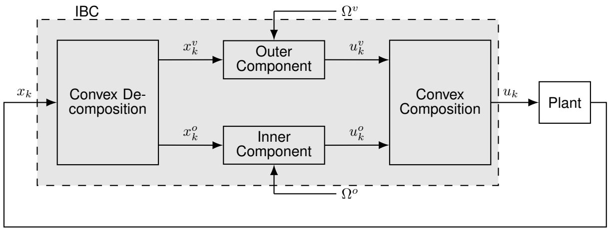

Figure 3: IBC block diagram

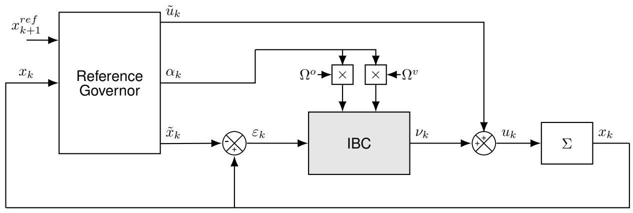

Figure 5: IBT block diagram

Practically, the inner set Ω^o is chosen as the same terminal maximal admissible invariant set Ω of MPC where the control law is an aggressive linear feedback law and the outer set is the maximal controllable set i.e the largest set where it exists a contol action leading any state to the origin.

The main contribution of this paper is to propose a preliminary study of an adaptation of IBC to the reference tracking under constraints to be applied to motion cueing framework.

3. Interpolation Based Tracking for MCA

The Interpolation-Based technique has been extended to the tracking problem in [Soy20] and the principle can be described as follow :

1. At step k, the optimization problem (7) finds a scaling factor $\alpha(k)$ and a virtual action $\tilde{u}(k)$ that moves a virtual state $\tilde{x}(k|k)$ in a neighborhood of the measured state $x(k)$ in Ω^v to minimize the cost function. This step is illustrated in Figure 4.

$$\begin{aligned} \underset{(\tilde{u}(k),\alpha(k))}{\text{minimize}} \quad & \|\tilde{x}(k+1|k)\|_{Q_x}^2 + \|u_{ref}(k) - \tilde{u}(k)\|_R^2 \\ \text{subject to} \quad & \tilde{x}(k+1|k) = A\tilde{x}(k|k) + B\tilde{u}(k) \\ & x(k) \in \{\tilde{x}(k|k)\} \oplus \alpha_k \Omega^v \\ & \{\tilde{x}(k|k)\} \oplus \alpha_k \Omega^v \subset \Omega^v \\ & \tilde{u}(k) \in (1-\alpha(k))\mathbb{U} \\ & \{\tilde{x}(k+1|k)\} \oplus \alpha(k)\Omega^v \subset \Omega^v \end{aligned} \qquad (7)$$

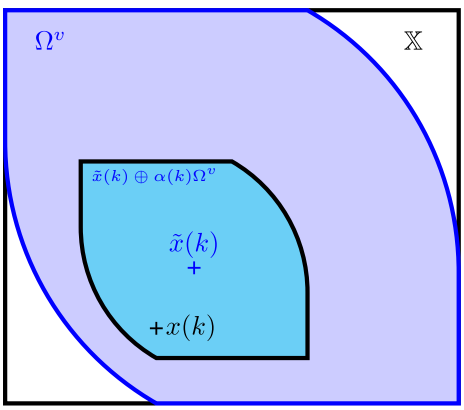

Figure 4: IBT principle

2. Then the state tracking error $\varepsilon_k = x_k - \tilde{x}_k$ is located in the scaled set $\alpha(k)\Omega^v$ and the IBC procedure of the previous section is applied in the scaled state space and control space.
3. The control action applied to the whole system can be computed as :

$$u(k) = \tilde{u}(k) + \nu(k) \qquad (8)$$

where $\nu(k)$ is the result of the IBC procedure on the scaled state space

The Interpolation-Based Tracking principle is summarized in Figure 5. In this procedure, only quadratic and linear programming algortihms are involved, consequently, implementation of IBT is close to a MPC one as it can be found in [Beg12], the main difference is the choice of outer and inner sets which are practically computed offline as convex polyhedrons.

4. Results and Discussion

In this section, we consider the lateral acceleration rendering during a slalom phase with respect to two motion cueing algorithms :

1. MPC-MCA with a prediction horizon of 3.2 seconds.
2. IBT-MCA with a convex polyhedral outer set Ω^v chosen as a controlled invariant set computed thanks to procedure in [Ngu11] and less complex as the largest controlled invariant set.

The considered driving simulator is Renault's ULTIMATE whose parameters are given in table 1.

Table 1: Parameters of ULTIMATE

Parameter	Notation	Limit
Position	p_k	2.6 m
Speed	v_k	3 m/s
Acceleration	a_k	5 m/s^2
Sampling Time	T_s	8ms

Figure 6 (resp Figure 7) depicts the state space trajectory of the system with respect to the IBT-MCA (resp MPC-MCA), The largest controlled invariant set C_N is represented in blue and the outer set Ω^v in red. The better exploitation of the state space of the IBT-MCA can be noticed

The comparison of acceleration renderings is depicted on Figure 8, during the slalom. The acceleration rendering of IBT appears to reproduce better the shape of the reference particularly during the period of variation change (from 4.5 to 5s).

The IBT-MCA has the advantage of not handling predictive states while better managing the exploitation

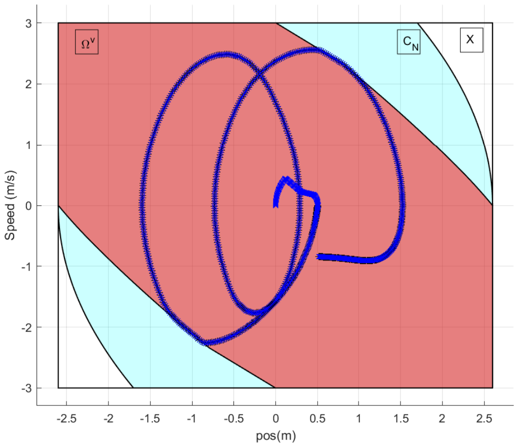

Figure 6: State-space trajectory for IBT

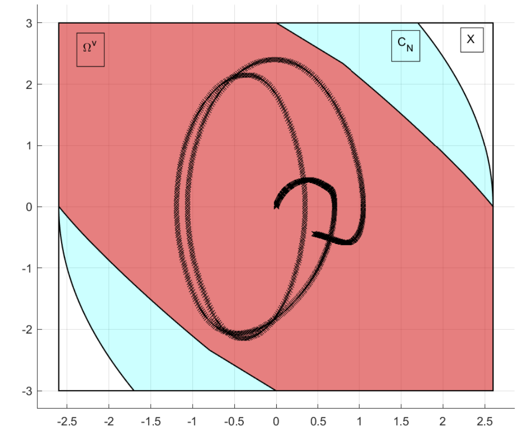

Figure 7: State-space trajectory for MPC

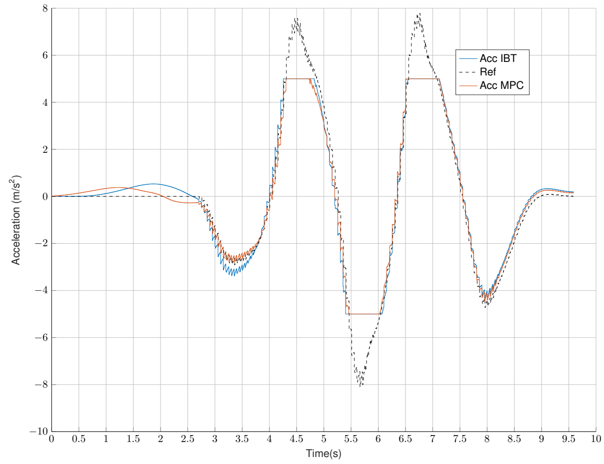

Figure 8: Comparison of acceleration rendering for IBT and MPC

of the state space by keeping the performance of long prediction horizon MPC-MCA. One can add the computation times for optimization solving are similar but need to be improved in further works by decreasing the complexity of .

Conclusion

MPC offers good renderings for position tracking when the trajectory is known but in the driving simulation framework, acceleration cannot be anticipated without strong hypothesis on future acceleration reference on a given prediction horizon. The presented method uses a short horizon and a better management of state feedback from a computational point of view and can overcome some limitations of real-time MPC such as the complexity induced by prediction and the multiplication of parameters to tune.

References

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), 6340–6345, IEEE, 2012.

M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier and N. Maïki, **MPC based motion cueing algorithm: Development and application to the ULTIMATE driving simulator**, in Conférence simulation de conduite, 221–233, 2004.

Z. Fang and A. Kemeny, **Motion cueing algorithms for a real-time automobile driving simulator**, in Driving Simulation Conference, Paris, France, Sept, 6–7, 2012.

Z. Fang and A. 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, 4–5, 2014.

Z. Fang and A. Kemeny, **An efficient Model Predictive Control-based motion cueing algorithm for the driving simulator**, *Simulation*, vol. 92(11): 1025–1033, 2016.

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*, vol. 50(1): 15965 – 15970, 2017, 20th IFAC World Congress.

S. Munir, M. Hovd, Z. Fang, S. Olaru and A. Kemeny, **Complexity reduction in motion cueing algorithm for the ULTIMATE driving simulator**, *IFAC-PapersOnLine*, vol. 50(1): 10729–10734, 2017.

H.-N. Nguyen, P.-O. Gutman, S. Olaru and M. Hovd, **Explicit constraint control based on interpolation techniques for time-varying and uncertain linear discrete-time systems**, *IFAC Proceedings Volumes*, vol. 44(1): 5741–5746, 2011.

H.-N. Nguyen, P.-O. Gutman, S. Olaru and M. Hovd, **Implicit improved vertex control for uncertain, time-varying linear discrete-time systems with state and control constraints**, *Automatica*, vol. 49(9): 2754–2759, 2013.

H.-N. Nguyen, S. Olaru, P. O. Gutman and M. Hovd, **Constrained control of uncertain, time-varying linear discrete-time systems subject to bounded disturbances**, *IEEE Transactions on Automatic Control*, vol. 60(3): 831–836, 2014.

C. Rengifo, J.-R. Chardonnet, H. Mohellebi, D. Paillot and A. Kemeny, **Feasibility Analysis For Constrained Model Predictive Control Based Motion Cueing Algorithm**, in 2019 International Conference on Robotics and Automation (ICRA), 2076–2082, IEEE, 2019.

M. Soyer, S. Olaru and Z. Fang, **Interpolation Based Control for reference tracking under constraints**, in European Control Conference 2020, 2020.

J. Venrooij, D. Cleij, M. Katliar, P. Pretto, H. Bülthoff, D. Steffen, F. Hoffmeyer and H. Schöner, **Comparison between filter-and optimization-based motion cueing in the Daimler Driving Simulator**, in Driving Simulation Conference, 2016.