Motion cueing algorithms for a real-time automobile driving simulator

Zhou FANG, Andras KEMENY

RENAULT, Technical Center for Simulation, TCR AVA 0 13, 1 avenue du Golf, 78288 Guyancourt, France

e-mail : zhou.fang@renault.com, andras.kemeny@renault.com

Abstract

The MCA (Motion Cueing Algorithm) for driving simulator takes into account the simulator's workspace limits and the driver's motion perception thresholds to reproduce simulated vehicle's accelerations. For the motion based driving simulators, the most applied MCAs are the classical and the optimal filters. This paper presents a new algorithm, called MPC (Model Predictive Control) explicit algorithm. Compared with the filters' algorithms, the MPC integrates directly the system constraints into its optimization process, and then gives a real optimal solution and hardly needs the tuning process to check the workspace limits and the driver's perception thresholds. The reported MCA studies based on MPC implicit algorithm need high computational costs which can destroy the stability properties of optimal MPC in a real-time system. The proposed workspace limit condition improves significantly the MPC optimal stable condition for its application in MCA. The current MPC algorithm can achieve the complicated 2dofs optimization.

Key words: *Motion cueing algorithm, explicit MPC algorithm, washout filter, tilt technique, real-time simulator.*

I. Review of motion cueing algorithms

A. Classical filter algorithm

The classical filter is a rapid prototype method to develop motion cueing algorithm. By taking the road information as workspace limit, the car's lateral dynamics in a straight road can be realistically reproduced in a driving simulator [Gra1, Fis1]. Using different 1st and 2nd order HP filter parameters, for the 8 dofs simulator, Fischer et al. [Fis1] proposed to decompose the linear acceleration signal into high, middle and low components, i.e. linear hexapod, linear x, y rails and tilt angle command signal. The idea is worth being studied. However, the actuators' performance limits and the different transfer functions between rail system and hexapod could make the tuning task very difficult without modern control theory. The authors have also introduced a lane-signal feedback loop which allows applying rather optimally the tilt coordination technique for lateral motion cues. This information is in fact very useful to choose the best strategy for MCA [Cha1]. We think that the way of taking into account road information and driver's attention in MCA is an interesting research issue in order to reduce the false cues. The table 1 summarizes the classical filters' roles and the corresponding conventional parameter settings.

Table 1: classical filters and corresponding parameter settings

Filter Type	Motion rendering for	Filter transfer function: H(s)	Conventional pulsation, ω_n, and damping ratio, ξ, values [Fis1, Ron1]	Final position to step input signal: $y(t \to \infty)$
1st order HP	Rotational rate	$\frac{k.s}{(s+\omega_b)}$	$\omega_b = 0.2$	$\frac{k}{\omega_b}$
2nd LP	Tilt technique	$\frac{k.\omega_{n2}}{(s^2+2\xi.\omega_{n2}.s+\omega_{n2}^2)}$	$\omega_{n2} = 0.65$~1.02, $\xi=1$ $\omega_{n2} = 1.2$~2.5, $\xi=1$	
2nd HP	Linear acc.	$\frac{k.s^2}{(s^2+2\xi.\omega_{n1}.s+\omega_{n1}^2)}$	$\omega_{n1} = 2.5$~4.0, $\xi=1$~1.4	$\frac{k}{\omega_{n1}^2}$
Washout	Linear acc.	$\frac{k.s^3}{(s^2+2\xi.\omega_{n1}.s+\omega_{n1}^2).(s+\omega_w)}$	$\omega_w = 0.1 \sim 0.5$	0

The filter's parameters should be designed with the worst case: a unit step acceleration whose corresponding position is: X(s) = $1/s^3$. The simulator final steady position, y(t) can be evaluated by:

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} s.H(s).X(s) = \lim_{s\to 0} \frac{ks^2}{\left(s^2 + 2\xi.\omega_{n1}.s + \omega_{n1}{}^2\right)s^2} = \frac{k}{\omega_{n1}{}^2} \qquad (1)$$

To avoid the workspace violation, the parameters should be determined by:

$k/\omega_{n1}^2 \le k_p.x_{max}$, if $\xi_1>1$, k_p=1 otherwise k_p<1and max(gain. $\ddot{x}(t)$) = k.

B. Adaptive filter algorithm

As the classical filter needs to be designed in the worst case, the available workspace is often badly used. Another issue is the false cues generated by this simple technique which can be considered as a main cause for the occurrence of motion sickness.

UTIAS and NASA [Tel1], Nahon and Reid [Nah1], suggested an adaptive washout algorithm with a similar formulation as the classical filter in the time domain. The parameters are self-tunable and determined by optimizing a quadratic cost function (the acceleration difference between simulated vehicle and platform within the performance limits). The proposed algorithm with tilt technique involves 8 weight tuning parameters for 2dofs optimization. The main issue of this algorithm is its stability and the difficulty to determine the best range of adaptive parameters P_{xi}. The non-significantly improved results given by this algorithm seem much compromised with the complexity of the calculation effort. We can also observe that the lack of tilt acceleration constraint could be critical for applying the tilt coordination technique in a vehicle driving simulator (see fig. 14). A LP filter or additional weight parameter for tilt acceleration seems necessary to limit the false cues level within the human motion perceptual threshold. The adaptive filter for 2dofs MCA proposed by UTIAS and NASA is given by [Tel1]:

$$\ddot{x}_s(t) = P_{x1}.\gamma_{xveh}(t) - K_{x1}.\dot{x}_s(t) - K_{x2}.x_s(t) \qquad (2)$$
$$\dot{\theta}_s = Lim\left(P_{x2}.\gamma_{xveh}(t)\right) + P_{x3}.\dot{\theta}_{veh}(t)$$

where P_{x1}, P_{x2} and P_{x3} are the adaptive parameters and K_{x1}, K_{x2} are equation constants. The P_{xi} and the cost function are determined by:

$$P_{xi} = -G_i \frac{\partial J_x}{\partial P_{xi}}$$

$$J_x = 0.5[(\gamma_{xveh} - \ddot{x}_s)^2 + W_{x1}\left(\dot{\theta}_{veh} - \dot{\theta}_s\right)^2 + \rho_x\left(W_{x2}.\dot{x}_s{}^2 + W_{x3}.x_s{}^2 + W_{x4}.\dot{\theta}_s{}^2 + W_{x5}.\theta_s{}^2\right) + W_{x6}\left(P_{x1} - P_{x10}\right)^2 + W_{x7}\left(P_{x2} - P_{x120}\right)^2 + W_{x8}\left(P_{x3} - P_{x30}\right)^2]$$

C. Optimal filter algorithm

The optimal filter taking into account models for vestibular system was proposed by Sivan et al. [Siv1]. This algorithm uses techniques of optimal control and minimizes the driver's perception error between the vehicle and the simulator. Chen and Fu [Che1] have studied the algorithm based on different techniques (optimal, fuzzy compensation) to find a better optimal solution. In the formulation of Chen [Che1] and Telban[Tel2], the input signal is [dθ/dt $u_{s\ lin}$]. It has been found impossible to control the tilt acceleration level without introducing $d^2\theta/dt^2$ into input signal. Nevertheless, as we will discuss in paragraph (§-II.E), the control of tilt acceleration level is very important for tilt technique in vehicle driving simulator. So in our future research, a modified state-space model, as reported in [Tel1], by using tilt acceleration instead of tilt rate as input signal will be proposed and analysed for explicit MPC algorithm.

D. Model Predictive Control algorithm

Dagdelen [Dag1] and Augusto [Aug1] have studied an implicit MPC algorithm for MCA application. The authors' works are based on on-line MPC optimization technique. Due to the real-time requirement (step time in milliseconds), this technique would be limited by the predictive horizon time and iterative number. Morari et al. [Mor1] indicated that a limit on the online computation time can destroy the stability properties of optimal MPC. The explicit MPC algorithm is developed to rise above this weakness.

Both implicit and explicit MPC algorithms are based on the same optimization principle. A more detail explication is introduced in next paragraph. Due to the direct integration of the simulator's limits into MPC optimization strategy, the MPC algorithm is considered better than any other motion cueing strategy [Dag1, Aug1]. Unlike the classical filter algorithm, the MPC algorithm is a non delayed algorithm.

II. Explicit MPC algorithm

A. Principle of explicit algorithm

The MPC algorithm aims at finding an optimal control law, based on minimizing (or maximizing) the value of a cost function, for the following state-space model:

$$\begin{aligned} X_k &= A.X_{k-1} + B.U_{k-1} \\ Y_k &= C.X_{k-1} \\ H_x.X + H_u U &\le K \end{aligned} \qquad (3)$$

where $A \in R^{nxn}$, $B \in R^{mxn}$, $C \in R^{pxn}$. $U_k \in R^m$, $X_k \in R^n$, $Y_k \in R^p$ are the input, state and output vectors respectively. The constraints given by $H_x \in R^{qxn}$, $H_u \in R^{qxm}$, $K \in R^q$ define a polyhedral region containing the origin in their interior, and the pair (A,B) is assumed stabilizable.

The optimal control law for current state, X_k, can be obtained by means of multi-parameter programming algorithm with a quadratic cost function defined by:

$$J_N^*(X_k) = \min_{u_0, u_1, \ldots, u_{N-1}} \|X_N\|_{Q_N}^2 + \sum_{i=0}^{N-1} \|U_i\|_R^2 + \sum_{i=0}^{N-1} \|X_i\|_Q^2 \qquad (4)$$

subj. to: $X_i = A.X_{i-1} + B.U_{i-1}$, $X_0 = X_k$, $1 \le i \le N$, $X_N \in T_{set}$ $H_x.X_i + H_y.U_i \le K$, $0 \le i \le N-1$

with weighting matrices $R > 0$, $Q \ge 0$, $Q_N \ge 0$ and $(Q^{1/2},A)$ is detectable.

At each sampling time, the current state X_k is used to find an open-loop control sequence: $K_{opt\ k} = [u^T_{0|k}, u^T_{1|k}, \ldots, u^T_{N-1|k}]^T$ where only the first element is applied for the plant system control and the procedure is repeated by following a "receding horizon control" principle. Thereby, the plant state evolution can be considered as a result of close-loop system. The future reference trajectory, updated at each moment, *t*, is generally supposed as constant (cf. fig. 1).

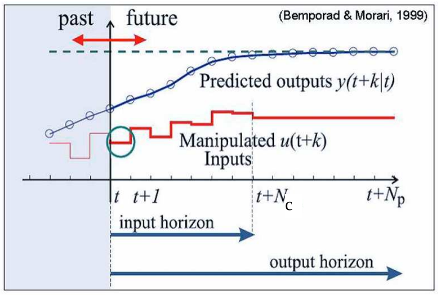


Fig. 1: Principle of MPC algorithm [Bem1]

For the explicit MPC with multi-parameter Quadratic Programming (QP) algorithm, the optimal control law, $U^*_0(X)$, can be deduced from an affine function [Kva1]:

$$U_0^*(X) = F_i.X + G_i, \text{ for } X \in R_i, \qquad (5)$$

where R_i is a polyhedral region in R^n.

The parameters F_i and G_i are given by the MPT (Multi-Parametric Toolbox) developed by the Automatic Control Laboratory of ETH, Zürich (http://control.ee.ethz.ch/~mpt/) for Matlab. The principle of explicit algorithm is recited as follows: By multi-parametric programming, a linear or quadratic optimization problem is solved off-line. The associated solution takes the form of a piecewise affine state feedback law. In particular, the state-space is partitioned into polyhedral sets and for each of those sets the optimal control law is given as one affine function of the state.

The main on-line computational time is consumed for finding the appropriated region by the test: $|H_i.X(k) – K_i|$ < numerical error tol. (e.g. = 10^{-7}). Hence the maximal computation cycle time can be evaluated by the test over all regions during MPC algorithm design phase.

B. Reference Tracking model

Compared with the regulation problem where the reference state is fixed, the reference tracking optimization is to treat a time varying reference problem. The reference signal, r, is described by following model [Pek1]:

$$z_{k+1} = A_r.z_k$$
$$r_k = C_r.z_k \tag{6}$$

where: $r_{min} \leq r_k \leq r_{max}$

We can define an augmented state space model in order to transform the optimization tracking problem to a standard regulation optimization:

$$\begin{aligned} \hat{X}_{k+1} &= A_m.\hat{X}_k + B_m.u_k \\ \hat{Y}_k &= C_m\hat{X}_k + D_m.u_k \end{aligned} \tag{7}$$

where:

$$\hat{X}_k = \begin{bmatrix} X_k \\ z_k \end{bmatrix}, \quad A_m = \begin{bmatrix} A & 0 \\ 0 & A_r \end{bmatrix}, \quad B_m = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad C_m = \begin{bmatrix} C & -C_r \end{bmatrix}, \quad D_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

In driving simulator with A_r=1, C_r=1, the tracking reference is the acceleration signal for the linear acceleration rendering or the angular position for the tilt coordination technique.

C. Algorithm stable condition

Bemporad and Morari et al. [Bem1, Bem2], have summarized different stability conditions for MPC algorithm of which we briefly review those related to the present study. The stability conditions can be classified into two categories: using a cost function to minimize the command sequence at each moment *t* as a Lyapunov function and those shrinking the state in some norm to guarantee future feasible solution:

- End constraint is imposed by forcing the predictive end state to equal its origin,
- Invariant terminal set (the idea is to relax terminal constraints into set-membership constraints, where the feedback gain, K_{LQ}, corresponds to the invariant set under LQ regulation and the constraints are fulfilled inside it).

For motion cueing algorithm, the second condition is more profitable, because it gives a more efficient exploitation of simulator workspace for the same predictive horizon steps. Thanks to the work of different automatic experts [Bem1] and MPT group, an algorithm determining automatically the LQ terminal set is implemented in MPT. For MCA and 1dof optimization, this technique can provide an appropriated real-time solution. But for 2dofs optimization, it's difficult to achieve a convergent solution, which is the reason why this paper developed a new special MCA invariant set to overcome the encountered difficulty.

Our stable condition is based on the following assumption:

If the simulator can provide a large enough available workspace, it performs the acceleration or angle tracking task. When the simulator approaches its workspace limit, the simulator should be slowed down to reach its saturated position or to be turned back to some fixed point (washout process). To apply the washout process or not depends on the simulated scenario. For instance, the former one is more suitable for slalom profile test, whilst the washout is more adapted for a free driving test.

For the convenience of demonstration, we first suppose that the simulator approaches its limit with a constant deceleration. The workspace restriction can be described by:

$$|x_i + v_i.T + u_i.T^2/2| \le x_{max} \qquad (8)$$

To avoid any violation of the simulator workspace boundaries and to keep a non conflicting perceived acceleration, it is necessary in terminal braking stage to pass a zero velocity state. Thereby:

$$u_i = -v_i/T \quad \text{with } u_i \le u_{thd} \qquad (9)$$

which leads to the following relationship between T and u_{thd} :

$$T = \sqrt{\frac{2(x_{max} - x_i)}{|u_{thd}|}} < 2\sqrt{\frac{x_{max}}{|u_{thd}|}} \qquad (10)$$

Supposing x_{max} = 2.6m and x_i = 0, u_{thd} = 0.2m/s², it produced: T_{max} = 5.1s

Theoretically, the T_{max} is the minimal horizon (for a constant horizon) to explore the optimal solution in the whole workspace range. Even so, such horizon time remains too large to be applied in a real time system in practice. Consequently, other conditions such as terminal invariant set have to be used to guarantee the stability condition. In the following paragraphs, is proposed a new stable condition particularly efficient for MCA which can guarantee the stability condition even with a short horizon. The concept is to add an appropriated workspace limit condition instead of imposing terminal invariant set in the MPC optimization process.

In the MPC algorithm, the relation (8) is checked at each sampling time step. For an open-loop, it describes the condition $x(t+T) \le x_{max}$. But MPC algorithm actually gives a close-loop optimal result due to its feedback state information. During braking phase of motion cues, i.e., (8) lies at its equality restriction, the input acceleration, u, should be written as a time-variant signal as well as the simulator state:

$$x_i(t) + v_i(t).T + u_i(t).T^2/2 = x_{max} \qquad (11)$$

Eq. (11) is an unstable condition for simulator, because the temporal signal x(t) can exceed the limit value x_{max} before reaching its final steady value x_{max} which will be proved later. Concerning the stability of equation (11), we introduce a modified condition:

$$x_i(t) + c_v.v_i(t).T + c_u.u_{thd}(t).T^2/2 = x_{max} \qquad (12)$$

whose corresponding Laplace function is :

$$x_i(s) = \frac{\left[s^2.(x_0/x_{max}) + s.(v_0 + 2\xi.\omega_n.x_0)/x_{max} + \omega_n^{\ 2}\right]x_{max}}{s.\left(s^2 + 2\xi.\omega_n.s + \omega_n^{\ 2}\right)} \qquad (13)$$

This is a typical second order time-invariant system with the step input x_{max}/s, system natural frequency $\omega_n = [2/(c_u.T^2)]^{0.5}$ and damping ratio: $\xi = c_v/(2\ c_u)^{0.5}$, x_0 and v_0 constitute the last tracking vehicle state from which the braking process slows down or the washout process starts.

The steady value can be evaluated by:

$$\lim_{t\to\infty} x(t) = \lim_{s\to 0} s.x_i(s) = x_{max}$$

Naturally, the $v(t\to\infty) = 0$.

According to the second-order system (13), if $\xi<1$, the oscillation in response of x(t) around x_{max} appears. Hence, the condition (11) is not a stable condition. To avoid the overshoot phenomenon during the future coming time, the necessary and sufficient condition is: $c_v^2/(2c_u) \ge 1$.

Like some classical filters, it's possible for the MPC to take $\xi<1$, but the final steady position should be reduced in consequence.

The evolution of acceleration is given by the following formulation:

$$u_i(t) = A_1.\exp(-k_1.\xi.\omega_n t) - A_2.\exp(-k_2.\xi.\omega_n t) \quad (14)$$

$$\textit{with} \quad A_1 = \frac{\omega_n(v_0 - 2k_1.v_0.\xi^2 - k_1.x_0.\xi.\omega_n + k_1.\xi.\omega_n.x_{max})}{\xi(k_1 - k_2)}$$

$$A_2 = \frac{\omega_n(v_0 - 2k_2.v_0.\xi^2 - k_2.x_0.\xi.\omega_n + k_2.\xi.\omega_n.x_{max})}{\xi(k_1 - k_2)}$$

$$k_1 = 1 - \sqrt{1 - \xi^{-2}}, \qquad k_2 = 1 + \sqrt{1 - \xi^{-2}}$$

Note that the simulator's restriction on velocity is not yet taken into account in above analysis. An exponential asymptotic law can be adopted to restrict the maximal velocity: $v(t) = (v_0 - v_{max}).\exp(-t/T_v) + v_{max}$ or in term of its corresponding acceleration limit: $u_{lim\ v}(t) = [v(t) - v_{max}]/T_v$. Finally, the max{$u_i(t)$, $u_{lim\ v}$, $-u_{lim\ perf}$} or min{ $u_i(t)$, $-u_{lim\ v}$ $u_{lim\ perf}$} should be sent as simulator input signal.

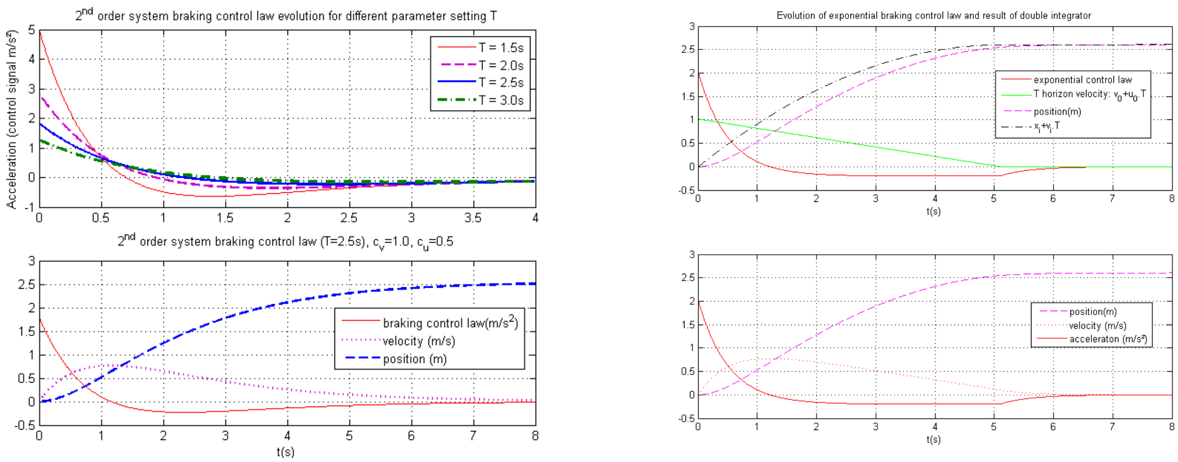


Fig. 2 : Simulator state evolution after x_0=0m v_0=0m/s and $u_s(t_0)$=2m/s² (left for eq. (14), right for table 2)

The tracking process associated with the developed braking law (eq.-12~14) can be considered as a special adaptive filter. Actually, the equation (12) is an invariant set and a workspace limit condition for MPC used only if the relation (12) is fulfilled. Here come the questions like whether it is possible to keep the braking deceleration as a constant value, e.g. equal to u_{thd}? In fact, such control law can result from a new assumption as follows:

$$u_i(t) = (u_0 + u_{thd}).\exp(-t/T_a) - u_{thd} \quad (15)$$

The deduced control laws are in fact very simple. They are summarized in table 2 and shown in the right of fig.2.

Table 2: constant u_{thd} washout law for ideal simulator

	if $v_i + u_i.T \geq 0$	else ($v_i + u_i.T < 0$)
v_i>0	$u_{lim\,x\,sup} = \frac{\sqrt{2u_{thd}(x_{max} - x_i - v_i.T)} - v_i}{T}$	$u_{lim\ x\ sup}$= $-v_0/T$
v_i<0	$u_{lim\ x\ inf}$= $-v_0/T$	$u_{lim\,x\,inf} = \frac{-\sqrt{2u_{thd}(x_{max} + x_i + v_i.T)} - v_i}{T}$

Although this proposed braking law (15) cannot be directly used as the MPC explicit algorithm workspace limits condition, this control law can be efficiently applied to 1dof MCA application, if the state-space model is an ideal simulator (double integrator).

D. State space model of driving simulator

For a cost-effective MCA, using the simulated plant to represent a real disturbed system is preferable, because a non symmetric driving scenario such as braking or mountain road lasts hardly over 15mn. At the end of a test of such duration, the simulated position is very close to simulator's real position. Actually, non symmetric scenario can produce more derivation than a symmetric maneuver e.g. sinus if one takes a first order system transfer function [Fan1] for the simulator. The benefit of using non disturbed system lies in the fact that the control law acts on acceleration, u, rather than on its differential term, du which is normally used to avoid the offset-free tracking. Thus, we can reduce the state-space equation dimension, which is important in order to develop explicit MPC real-time algorithm and to facilitate parameters' tuning.

In the current paper, the simulator is supposed to be an ideal one, i.e., the actuator manufactory controller is perfect. The figure 3 illustrates the scheme of MPC algorithm integration in the simulator.

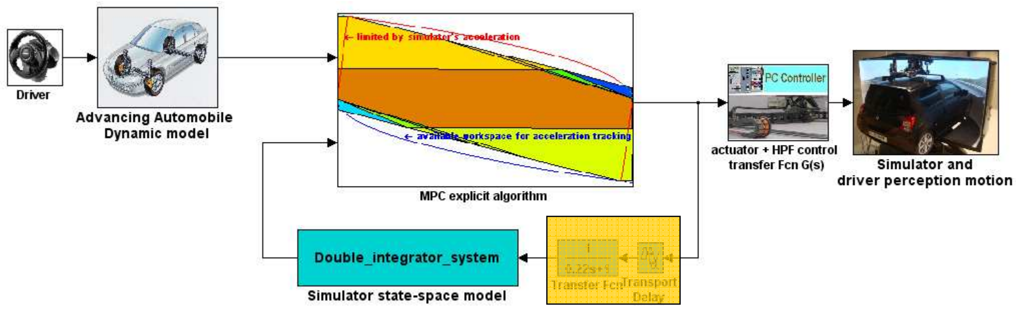


Fig. 3: Simulator model and MPC algorithm integration

The ideal simulator corresponds to a double integrator. With the proposed condition (12), the equation for MPC algorithm optimization can be formulated by:

$$\hat{X}_k = \begin{bmatrix} x_k \\ v_k \\ \theta_k \\ \omega_k \\ r_k \end{bmatrix}, \quad A_m = \begin{bmatrix} 1 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_m = \begin{bmatrix} 0.5dt^2 & 0 \\ dt & 0 \\ 0 & 0.5dt^2 \\ 0 & dt \\ 0 & 0 \end{bmatrix}$$

$$C_m = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & sign.g & 0 & -1 \end{bmatrix}, \quad D_m = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad U_m = \begin{bmatrix} \dot{v}_k \\ \dot{\omega}_k \end{bmatrix} \tag{16}$$

$$sign = 1,\ for..x..and..sign = -1..for..y[].$$

$$J_N^*(\hat{X}_k) = \min_{u_0,u_1,\ldots,u_{N-1}} \left\|\hat{X}_N\right\|_{\hat{Q}_N}^2 + \sum_{i=0}^{N-1} \left\|u_i\right\|_R^2 + \left\|x_i\right\|_{\hat{Q}x}^2 + \sum_{i=0}^{N-1} \left\|(u_i - r_i)\right\|_{Qu}^2$$

$$subject \quad to:$$

$$\left| x_k + T_1.v_k + \frac{T_1^2}{2}.u_k \right| < L_{x,y\lim}/2 \qquad |x_k| \le x_{\max} \quad |v_k| \le v_{\max} \quad |\dot{v}_k| \le \dot{v}_{\max}$$

$$\left| \theta_k + T_1.\omega_k + \frac{T_2^{\ 2}}{2}.d\omega_k/dt \right| < \theta_{pitch,roll\lim}/2 \quad |\theta_k| \le \theta_{\max} \quad |\omega_k| \le \omega_{\max} \quad |\dot{\omega}_k| \le \dot{\omega}_{\max}$$

E. Motion perception threshold

As mentioned in published results of Max Planck Institute [Bey1], human self-motion perception involves the contribution of different sensory systems and central mechanisms: the visual, vestibular, tactile and kinesthetic sensors along with brainstem and higher cortical processing. One of the interesting results reported is that the linear acceleration perception threshold can be described by a power law (cf. figure 4).

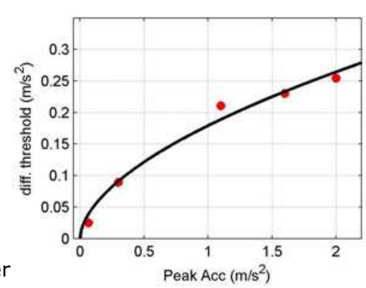


Fig. 4: Differential threshold for vertical translation increases with stimulus intensity in accordance with Steven' power law (from [Bey1])

According to the result of figure 4, the linear motion perceptual threshold increases with the acceleration level. Compared with our braking law (14), this allows us to choose the best T parameter based on the simulator performance and simulation requirements. The figure 5 gives an example for tuning T and u_{thd}.

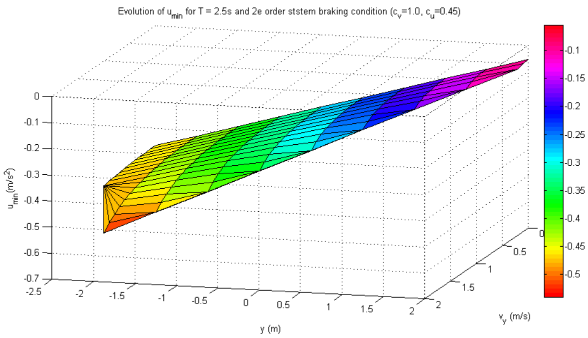


Fig. 5: Evolution of u_{min} from (14) in function of lateral position, y_0, and velocity, v_{y0}, for T = 2.5s and c_v=1, c_u=0.45

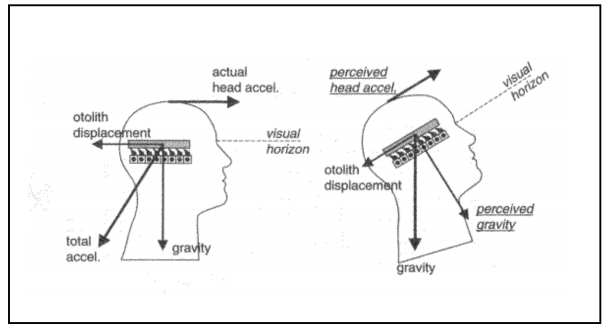


Fig. 6: Tilt coordination principle (from [Rey1])
$\gamma_{tilt} = g.\sin\theta \cong \gamma.\theta$ for $|\theta| < 0.2$ rad

Although the rail system can provide a satisfied acceleration level for slalom or sinus test, for rendering a sustained acceleration, the tilt coordination technique is necessary. This technique is based on the fact that the vestibular system cannot distinguish between inertia force produced by a linear acceleration and the effect of gravity (cf. fig. 6).

Distortions of the subjective vertical sensation may occur when the tilt angle exceed 20~30° (Aubert effect) [Rey1], therefore sustained acceleration simulated by tilt coordination should not exceed 0.5 g.

Since the driver is very sensitive to the tilt rate or the tilt acceleration [Rey1], the tilt rate should be within driver's perception threshold. Generally, it is found that the tilt rate is about 2~4°/s. However, in a driving simulator, this value is too limited to reproduce a realistic driving simulation. In practice, the tilt detection thresholds may be significantly raised above the theoretical level. For example, the PSA subjective tests at VTI simulator [Cha1] have provided higher tilt rate thresholds which increase with the linear motion level and are given in the following table:

Table 3: parameter settings From PSA paper [Cha1]

Perception threshold	$\gamma_{x,y}$ **linear acceleration**	
	0 m/s²	**1m/s²**
Linear acceleration	0,15m/s²	0,15m/s²
Tilt rate	2°/s	6°/s
Tilt acceleration	8°/s²	11°/s²

In a subjective ratings study of a 6dofs simulator, Fisher et al. [Fis2] have found that, for an emergency braking maneuver, the participants ratings of realism were better in high tilt rate (30°/s) condition than in low tilt rate (3°/s) one.

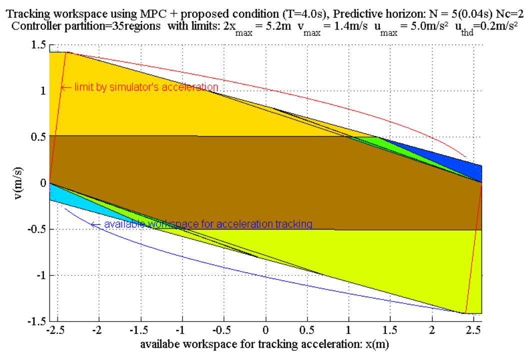
For 8dofs simulator, it's better to perform with the hexapod only the tilt motion, given that the system delay between rail and hexapod could be different, e.g., Renault's simulator ULTIMATE has 200ms for rail system and 80ms for hexapod at 0,2Hz [Fan1]. To exploit completely the performances of hexapod and rail systems for rendering a linear acceleration, we have to integrate the different actuator systems in the state-space model. This is a new future subject that we can treat with the MPC technique. The tilt rotation axe based on the head position can produce a close equivalent acceleration due to the human's tilt-translation ambiguity. But this rotational motion will also generate a linear motion at the seat position. The driver can detect this acceleration, given by $h.d^2\theta/dt^2$, if its level is high enough, where h is the distance between head and seat and θ, the tilt angle. For tilt acceleration threshold of 11°/s², its corresponding linear acceleration in seat is about to 0,2m/s² (with h=1m e.g.). This value may be found by chance in accordance with vestibule or self-motion detection thresholds. As the first tuning parameters, we have fixed: $|\theta| < 12°$, $|d\theta/dt| < 6°/s$; $|d^2\theta/dt^2| < 11°/s^2$.

Some special constraints can be added to adapt the driver perception threshold with linear acceleration level.

III. Simulation and experimental results

A. 1 DOF simulation result

Figure 7 illustrates the workspace exploited by MPC algorithm using proposed invariant set with N=5 and Nc=2. With a bigger number N (>200), the available workspace can be almost covered. Whilst with the standard LQ invariant set method provided by MPT, the N should be superior to 500 to reach an acceptable workspace (see fig. 8).



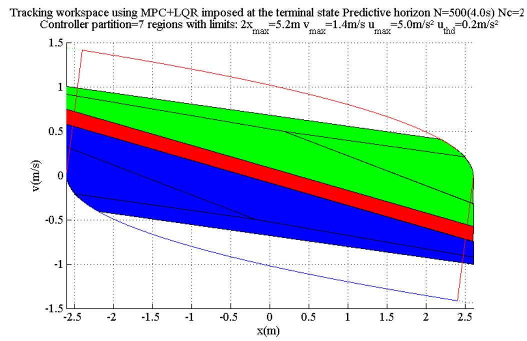


Fig. 7: Workspace used by MPC algorithm based on proposed law (N=5 and Nc=2)

Fig. 8: Workspace covered by MPC algorithm based on standard LQ invariant set (N=500 and Nc=2)

Figure 9 shows an example of motion cues for lateral acceleration given by MPC algorithm. The reference signal measured from slalom driving test is quite well restituted.

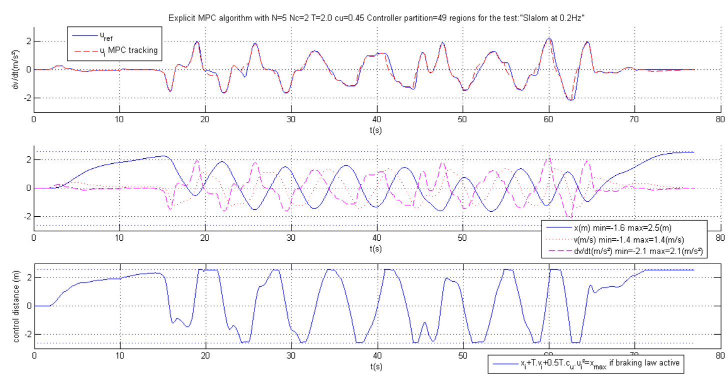


Figure 9: Illustration of lateral acceleration rendering for slalom test (top: rendering acceleration; middle: simulator state; bottom: braking control law is active if blue line is statured)

Fig. 10 illustrates the motion cues given by the explicit MPC algorithm designed for different parameter settings T.

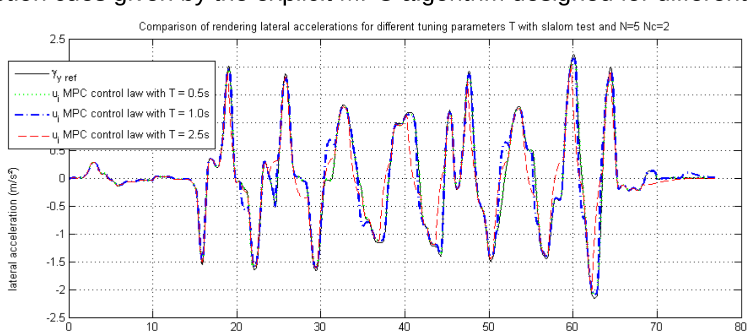


Fig. 10: Slalom test motion cues given by different parameter settings, T

For the current test, the rendering acceleration, given by the parameter setting T=0.5s, is superposed with the simulated car one. For the rendering result obtained with T=1.0s, 2~3 false conflicting cues are observed. In the case of the algorithm designed with T=2.5s, the braking phases appear more frequently, but the big conflicting false cues are reduced. Generally, the lower the value of T is, the lower the error between the restituted and the simulated acceleration is, but also the riskier it is to have big false cues.

Figure 11 displays a conventional tracking result: the pursuit of tilt angle. The reference signal from the paper [Cha1, PSA] corresponds to a lateral acceleration. It results from a double lane change maneuver followed by a 100m radius curve with vehicle speed of about 70km/h. The scenario is difficult to render with a universal algorithm without road information. The original authors have done a theoretical analysis through non delay classical filter (off-line simulation) and proposed an algorithm switching to render the first part of high-frequencies signal by rail's linear acceleration, whereas the second part by the combination of the tilt technique with the linear acceleration. We can find that the classical filter and the optimal filter provide very similar rendering results and have important delay. The MPC algorithm can provide non delay rendering result. Note that for the rapid acceleration signal, the false cues are also evident for the MPC algorithm owing to the tilt rate limit. However, as LP and HP filters' combination method, these false cues can be removed by using the linear rendering MPC algorithm for high frequencies signal and the tilt technique MPC algorithm for remaining signal. But it should be more reasonable to perform the motion cueing optimization with 2dofs motions at the same time. The next paragraph will discuss this issue.

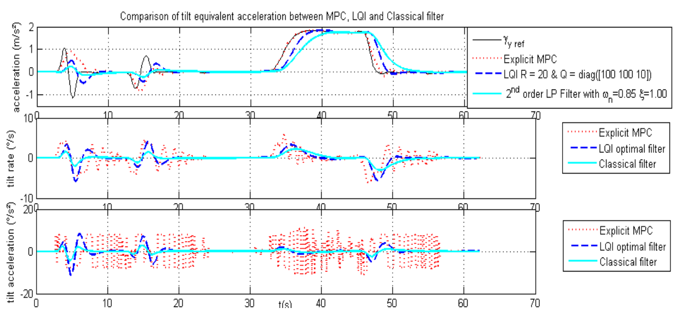


Fig. 11: Tilt equivalent acceleration from MPC, LQI and classical filter

B. 2 DOF simulation results

The motion cues for vehicle's x,y,z linear accelerations associated with corresponding rotation motions can be treated independently with simulator's corresponding linear motion and tilt rotation, here called 2dofs optimization for which a corresponding MPC explicit algorithm is developed. Using the same reference signal of PSA mentioned in the above paragraph, we compared the performances of classical filter, adaptive filter, optimal filter, and explicit MPC algorithm.

The classical and the optimal filters have been tuned only for the concerned reference signal, thus leading to an ideal theoretical result. They have not been validated for a free driving test. On the contrary, the explicit MPC algorithm is a general algorithm, applicable for free driving tests.

The classical filter gives a bad motion cueing result with some distortion phase for the first part and relatively important phase delay for the second part (cf. fig. 12). The UTIAS adaptive filter uses a cost function without tilt acceleration limit. According to the simulated result (see fig. 13), it is necessary to introduce this constraint. The rendering result with this technique is similar to that of classical filter for the first part of high frequencies signal and is better than the classical filter for the second part of low frequencies signal. But this conclusion may be wrong if the tilt acceleration constraint is taken into account.

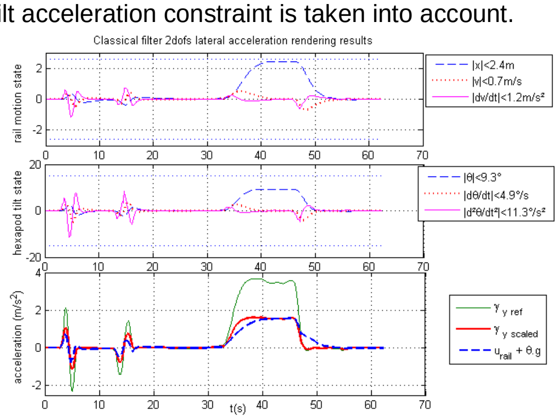


Fig. 12: Classical filter lateral acceleration rendering results

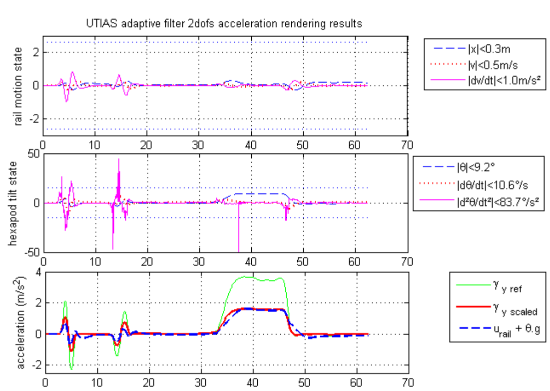


Fig. 13: Adaptive filter acc_y rendering results

The optimal filter gives an interesting result (see fig.14). But the tilt acceleration level is not acceptable. It's difficult and probably impossible to find the appropriated weight matrices to respect the tilt acceleration limit without it in the input signal. The figure 15 presents explicit MPC algorithm acceleration rendering results for which the tilt acceleration is actually controlled under the human perceived threshold limit. The result is better than the optimal filter one.

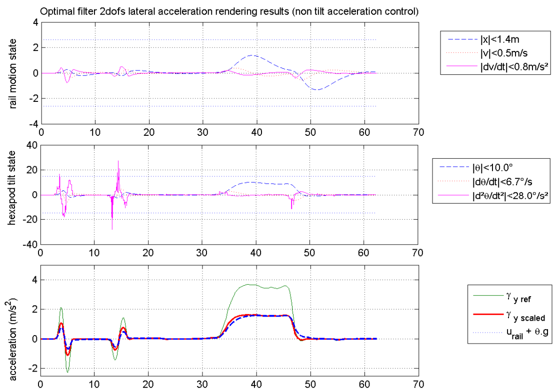


Fig. 14: Optimal filter 2dof γ_y rendering results

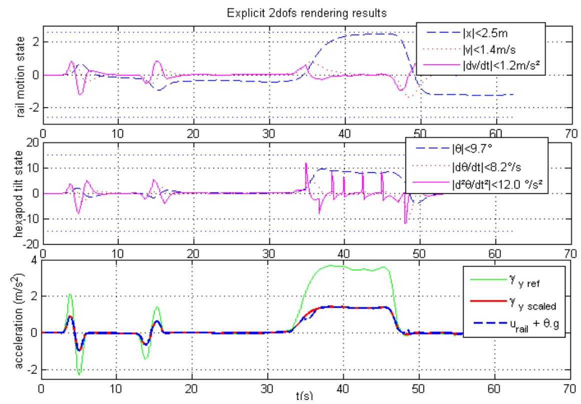


Fig. 15: Explicit MPC 2dof γ_y rendering results

In practice, we can adapt the algorithm for each scenario, e.g., using the pre-position technique like that analysed by Chapron et al. [Cha1] to optimize the simulator's workspace. This technique can be easily treated in the cost function. This corresponds to the tuning process task, not discussed here.

C. Driving test results

The first tuning of the MPC algorithm has been evaluated with an unlimited tilt acceleration control. The effect of tilt acceleration is obviously perceived by the driver and measured by CrossBow inertial system, fixed under platform. The weight coefficient for tilt rotation is not optimized. The second tuning MPC algorithm is prototyped by tilt acceleration control, i.e. $|d^2\theta/dt^2| < 11°/s$, $|d\theta/dt| < 8°/s$, $|\theta| < 12°/s$. The weight coefficients for linear acceleration input signal and for tilt rotation are modified. The effect owing to the tilt acceleration is considerably reduced. The driver feeling is improved. The evaluation of different tunings is planned in our future work.

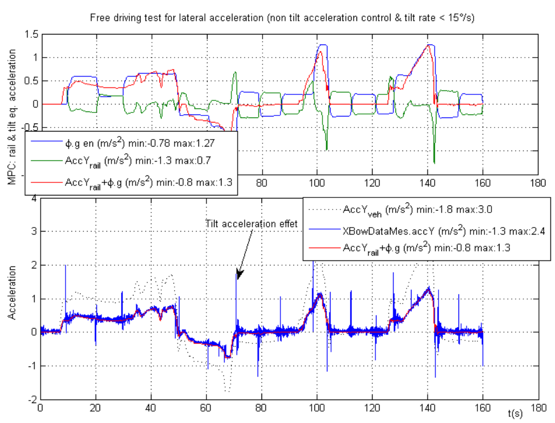


Fig. 15: Free driving test for sustained γ_y
(bad tuning parameters & high tilt acc. threshold: $|d^2\theta/dt^2|<60°/s^2$)

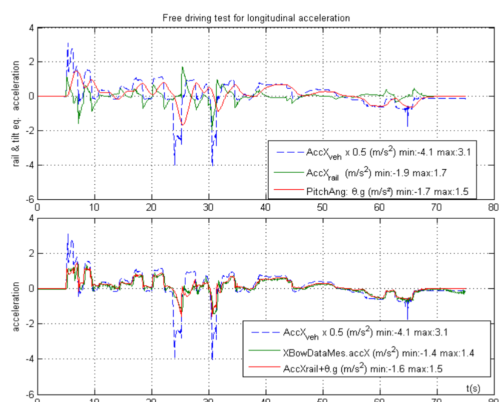


Fig. 16: Free driving test for γ_x
(improved tuning parameters)

IV. Conclusions

This paper presents a novel algorithm, called MPC explicit algorithm. An appropriated workspace limit condition, or in other words, a special adaptive filter is proposed, which can achieve the complicated 2dofs optimization by MPC explicit algorithm, so it reveals a new way to develop future MPC motion cueing algorithm. In comparison with the MCA used in flight simulators by NASA, Thales, TNO (optimal filter or in-line optimal filter) or in driving simulator (VTI, PSA, TNO, Ford, NADS etc., classical filter and optimal filter), the MPC algorithm possesses one important advantage: the ability of integrating the simulator's performances and the driver's perception threshold constraints into its optimization process. In such way, real optimal motion cueing results can be reached. Compared with classical filter method, the MPC algorithm can significantly reduce the distortion phase and the delay. In terms of tuning, it is also much easier than with the optimal method.

The first driving tests showed an interesting result. In combination with other information such as road situation and driver's attention, we can still improve the performance of the algorithm. We will prepare a series of test scenarios to evaluate the performance of the proposed algorithm and the optimal filter.

The relatively high number regions for some complicated constraints of the algorithm may provoke high computational costs, although the calculation formulation is very simple. Finding efficient algorithm to reduce polyhedral states-space, or using a parallel algorithm will promote this technique application in the field of driving simulator.

References

[Gra1] Grant P., Artz B., Blommer M., Cathey L., Greenberg J., "A paired comparison study of simulator motion drive algorithm" , DSC Europe 2002 Proceedings, Paris pp. 75-88

[Fis1] Fischer M., Sehammer H. and Palmkvist G., "Motion cueing for 3-, 6- and 8-degrees-of-freedom motion systems", DSC Europe 2010 proceedings, Paris, France

[Fis2] Fischer M. and J. Werneke, "The New Time-Variant Motion Cueing Algorithm For The DLR Dynamic Driving Simulator", DSC 2008, Monaco

[Cha1] Chapron T., Colinot J.P., "The new PSA Peugeot-Citroën Advanced Driving Simulator – Overall design and motion cue algorithm", DSC North America 2007 proceedings, Iowa City

[Ron1] Ronald A. Hess. ”Prediction of aircraft handling qualities using analytical models of the human pilot", J. NASATM84 233, 1982, 1-16

[Tel1] Telban Robert J., Weimin Wu, and Frank M. Cardullo, " Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms", NASA /CR-2000-209863

[Tel2] Telban Robert J. and Frank M. Cardullo, "Motion Cueing Algorithm Development: Human-Centered Linear and Nonlinear Approaches", NASA /CR-2005-213747

[Nah1] Nahon M.A., L.D. Reid and J. Kirdeikis, "Adaptive Simulator Motion Software with Supervisory Control", Journal of Guidance, Control, and Dynamics, vol.15, no.2 pp.376-383, 1992

[Siv1] Sivan R., Ish-Shalom J., and Huang J. K., "An Optimal Control Approach to the Design of Moving Flight Simulators", IEEE Transactions on Systems, Man, and Cybernetics, 1982. 12(6): p. 818-827

[Che1] Chen S.H., and Li-Chen Fu, "An Optimal Washout Filter Design with Fuzzy Compensation for a Motion Platform", 18th IFAC World Congress Milano, Italy August 28 - September 2, 2011

[Dag1] Dagdelen M., G. Reymond, A. Kemeny, M. Bordier, and NadiaMaýzi (2009). "Model-based predictive motion cueing strategy for vehicle driving simulators", Control Engineering Practice, Vol. 17(No. 9), 2009, pp. 995–1003

[Aug1] Augusto B.D.C., "Motion Cueing in the Chalmers Driving Simulator: An Optimization-Based Control Approach", Master of Science Thesis, 2009

[Kva1] Kvasnica M., "Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools", VDM Verlag (October 29, 2009)

[Pek1] Pekar J., V. Havlena, "Design and analysis of model predictive control using MPT ToolBox", http://dsp.vscht.cz/konference_matlab/matlab04/pekar.pdf, 2004

[Bem1] Bemporad A. and M. Morari, "Robust Predictive Control: A Survey", Robustness in Identification and Control, vol. 245, pp. 207-226, 1999

[Bem2] Bemporad A., M. Morari, V. Dua, E.N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems", Automatica 38(2002) 3-20

[Mor1] Morari M. & Jones C. et al., "Real-time Optimization for Distributed Model Predictive Control", 2010, Automatic control Laboratory, ETH Zürich

[Bey1] Beykirch K, Barnett-Cowan M, Zaichik L, Bos J, Ledegang W., "Human Motion Perception" http://www.kyb.tuebingen.mpg.de/research/dep/bu/motion-perception-in-vehicle-simulation/human-motion-perception.html

[Rey1] Reymond G, Kemeny A., "Motion Cueing in the Renault Driving Simulator", Vehicle System Dynamics, 34(2000), pp. 249-259

[Fan1] Fang Z., Reymond R., Kemeny A., "Performance identification and compensation of simulator motion cueing delays", DSC Europe 2010 proceedings, Paris, France