



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>
Handle ID: <http://hdl.handle.net/10985/7337>

To cite this version :

Baris AYKENT, Damien PAILLOT, Andras KEMENY, Frédéric MERIENNE - Influence of a new discrete-time LQR-based motion cueing on driving simulator - Optimal Control Applications and Methods - Vol. 35, p.DOI: 10.1002/oca.2081 - 2013

Any correspondence concerning this service should be sent to the repository

Administrator : scienceouverte@ensam.eu



Influence of a new discrete-time LQR-based motion cueing on driving simulator

B. Aykent^{1,*}, F. Merienne¹, D. Paillot¹ and A. Kemeny^{1,2}

¹CNRS Le2i Arts et Metiers ParisTech, Chalon sur Saone, France

²Technical Centre for Simulation, Renault, Guyancourt, France

SUMMARY

This study proposes a method and an experimental validation to analyze dynamics response of the simulator's cabin and platform with respect to the type of the control used in the hexapod driving simulator. In this article, two different forms of motion platform tracking control are performed as a classical motion cueing algorithm and a discrete-time linear quadratic regulator (LQR) motion cueing algorithm. For each situation, vehicle dynamics and motion platform level data are registered from the driving simulation software. In addition, the natural frequencies of the roll accelerations are obtained in real-time by using FFT. The data are denoised by using wavelet 1D transformation. The results show that by using discrete-time LQR algorithm, the roll acceleration amplitudes that correspond to the natural frequencies and the total roll jerk have decreased at the motion platform level. Also, the natural frequencies have increased reasonably by using the discrete LQR motion cueing (1.5–2.2 Hz) compared with using the classical algorithm (0.4–1.5 Hz) at the motion platform, which is an indicator of motion sickness incidence avoidance. The literature shows that lateral motion (roll, yaw, etc.) in the frequency range of 0.1–0.5 Hz induces motion sickness. Furthermore, using discrete-time LQR motion cueing algorithm has decreased the sensation error (motion platform–vehicle (cabin) levels) two times in terms of total roll jerk. In conclusion, discrete-time LQR motion cueing has reduced the simulator sickness more than the classical motion cueing algorithm depending on sensory cue conflict theory. Copyright © 2013 John Wiley & Sons, Ltd.

KEY WORDS: optimal control; linear quadratic regulator (LQR); discrete-time control; motion cueing; washout; driving simulator

1. INTRODUCTION

Multi-sensory fusion (visual, auditory, haptic, inertial, vestibular and neuromuscular) [1] plays important roles to represent a proper sensation in driving simulators. Thus, restituting the inertial cues on driving simulators is essential to acquire a more proper sensation of driving [2]. Because of the restricted workspace, it is not possible to represent the vehicle dynamics continuously with scale 1 to 1 on the motion platform. Nevertheless, the most desired aim is to minimize the deviation of the sensed accelerations between the represented dynamics as realistic as possible depending on the driving task. Therefore, washout filters are used to simulate the real-world vehicle dynamics into the realizable motion on the dynamic driving simulator (with a 6-DOF hexapod motion platform in this study). If the sensory cue conflict between vehicle and platform levels increases, it may cause simulator sickness. For that reason, simulator sickness has been one of the main research topics for the driving simulators. It was assessed between dynamic and static simulators [3,4]. However, there has been a very few publications of vehicle–platform cue conflict-based motion sickness depending

*Correspondence to: B. Aykent, Institut Image, CNRS Le2i Arts et Metiers ParisTech, 2 Rue T. Dumorey, 71100 Chalon sur Saone, France.

†E-mail: b.aykent@gmail.com

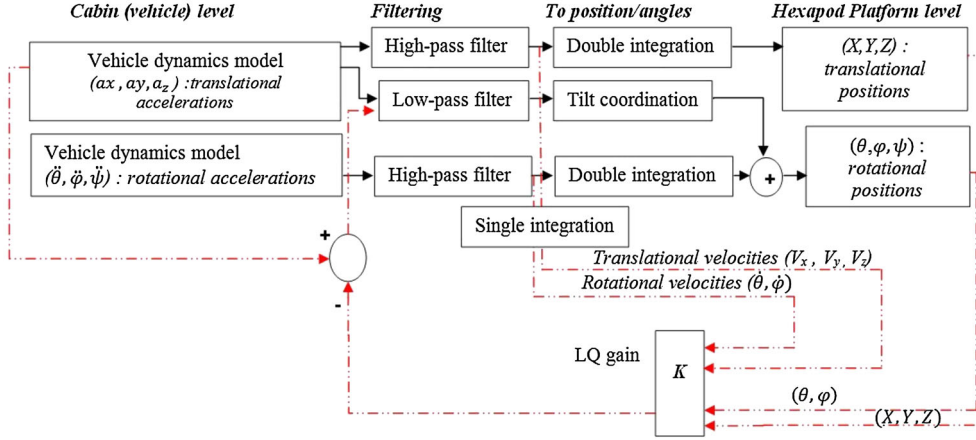


Figure 1. Discrete-time LQR-based motion cueing.

on natural frequencies, their corresponding roll acceleration amplitudes and the total sensed roll jerks. To reduce the simulator sickness, the difference between the accelerations through the visual (vehicle level) and inertial (platform level) cues have to be minimized (cost function minimization via discrete-time linear quadratic regulator (LQR), which is a certain type of optimal control, in this paper). Because of that fact, this paper addresses the simulator motion sickness as a function of vehicle–platform roll acceleration and roll jerk sensitivities. This work was performed under the dynamic operations of the Simulateur Automobile Arts et Metiers (SAAM) driving simulator as with a classical tracking control and a discrete-time LQR (DT LQR) tracking control. The dynamic simulators are being used since the mid-1960s (Stewart platform) [5] first for the flight simulators, and then the use has spread to the automotive applications. The utilization scope diversifies from driver training to research purposes such as vehicle dynamics control, advanced driving assistance systems [6] and motion and simulator sickness [7–9]. The dynamic driving simulator SAAM consists of a 6-DOF motion system. It acts around a Renault Twingo 2 cabin with the original control instruments (gas, brake pedals and steering wheel). The visual system is realized by an approximate 150 cylindrical view. Nearby the cabin, the multi-level measuring techniques (XSens motion tracker and Biopac electromyography (EMG) device) and postural stability analysis (Technoconcept platform) are available, which were already used with numerous attempts [7–9] such as sinus steer test, NATO chicane, sinusoidal roughness test and country road scenario. The visual (vehicle (cabin) level) accelerations of translations (longitudinal X , lateral Y and vertical Z axes) as well as the visual accelerations of roll and pitch, which correspond to the vehicle dynamics, were taken into account for the control. Then, the platform positions and velocities were controlled and fed back to the vehicle level in order to minimize the conflict between the vehicle and platform levels in this article (Figure 1). For the evaluation and validation procedures [4, 6–14], the scenario was driven on the simulator SAAM with a classical motion cueing and also with a constrained DT LQR controlled motion cueing to describe the impact of the feedback control. Some results from the simulator experiment were illustrated in the scope of this research with real-time controls of the platform at a longitudinal velocity of 60 km/h.

1.1. Problem statement

There are different components in the dynamic simulators, which are related to the control theory such as motion platform dynamics, vehicle (cabin) dynamics and human (muscular or head) dynamics. From those, only the vehicle and platform dynamics can be designed by researchers and engineers. The most important objective in dynamic simulators is to achieve the goal of consistent dynamics in those two levels. The human (muscular, head) dynamics are monitored depending on the changes of vehicle–platform dynamics interaction during the driving. That is why the human reaction dynamics is very difficult to study, because even if the same subject drives with the same

vehicle model, the same motion cueing algorithm and the same scenario, he or she will react in different ways. However, by using the closed-loop or open-loop tracking control of the hexapod platform, it can be possible to compare their performances with or without consideration of human factors.

There are also some evaluation methods for the driving simulation that also considers the human factors [7–9]. In [7], the effect of having adaptive motion cueing (a closed-loop tracking control of the motion platform) and classical motion cueing (an open-loop tracking control of the platform) algorithms was discussed in terms of the platform level dynamics and the subjective assessments regarding the simulator sickness. Because the pitch and roll oscillations were objectively decreased (measured and monitored via SCANERSTUDIO software) by using the adaptive motion cueing, the disorientation-related (dizziness, nausea) severity was subjectively rated (via simulator sickness questionnaires) lower compared with using the classical motion cueing [7]. Another possibility to evaluate the human factors is to use biofeedback methods [8, 9]. They are considered as an objective metrics when it is compared with the questionnaire method. However in both cases, the human factors are taken into account. The statistical analysis should be carried out in order to present the results precisely, because it could be possible to have convergence or divergence in terms of the biofeedback or the questionnaire analysis. In [8, 9], the neuromuscular analysis of the participants in the driving experiments was investigated. The vestibular (participants' heads) and arm muscle (EMG analysis) dynamics interactions were compared for the adaptive (closed-loop control of the platform) and classical motion cueing (open-loop control of the platform) algorithms [8] for the same driving scenario. Moreover, the subjective sickness was assessed via a questionnaire method [8]. The results revealed that the arm and vestibular dynamics cues conflict was decreased by using the adaptive motion cueing compared with using the classical one. The head–arm muscle dynamics interactions were discussed in [9] by using an LQR-based motion cueing (closed-loop tracking control of the platform) and a classical (open-loop tracking control of the platform) algorithm. The results from [9] indicated that using the LQR motion cueing provided an increased level of agility for the subjects in driving compared with using the classical motion cueing for the same driving scenario. In other words, the correlation of the head–arm muscles dynamics was yielded higher in LQR algorithm than the one in classical algorithm.

This study was accomplished by using an autopilot in order to have all the scenario settings identical such as steering, brake pedal, gas pedal and vehicle model (engine, axle kinematics, axle dynamics, etc.) without human factor. It surveyed the contribution of using a closed-loop tracking control, which is a DT LQR algorithm, of the hexapod platform on 'sensation error' in order to focus on the advantages associated to the cue conflict reduction. Here, the vehicle(cabin)–platform cues were compared in frequency domain by using two different types of motion cueing (DT LQR algorithm and classical algorithm).

This paper was organized as follows. Section 2 described the LQR-based motion cueing. Section 3 explained how the DT LQR control was developed and the data were processed. Section 4 continued with the results and discussion. Eventually, Section 5 concluded the paper.

2. LQR-BASED MOTION CUEING

In this section, the washout filter design methodology based on the LQR theory was briefly summarized, which constituted as the basics for the DT LQR motion cueing in Section 3. The main idea of the LQR-based washout filter design is to find a linear transfer matrix $W(s)$ that minimizes a certain quadratic cost involving the sensation error e and the simulator input u_s [7–9, 15–21] without allowing the hexapod platform to exceed the simulator constraints as shown in Figure 2. The LQR motion cueing algorithm design should be started with a mathematical model of the human vestibular system. For this study, the proposed input u to the vestibular system contains

$$u = \left[\int a_x dt \quad \int a_y dt \quad \int a_z dt \quad \dot{\theta} \quad \dot{\phi} \quad \dot{\psi} \right]^T$$

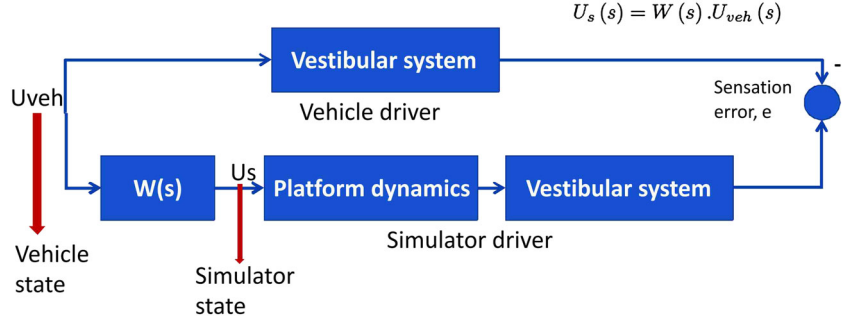


Figure 2. Optimal motion cueing algorithm.

As seen from this proposed input, it contains highly nonlinear elements such as translational and rotational velocities. In this article, the system is assumed linear by using Riccati equation. Then, the sensed rotational motion \hat{q} (pitch, roll and yaw velocities) can be given by the mathematical model of the semicircular canals [16, 20]. The sensed translational motion \hat{v} is associated to the input-specific force a by the otolith model [16, 20]. Denoting the vestibular system states from the vehicle and simulator drive as X_{veh} and X_s , the corresponding state error can be defined as X_e [20]:

$$X_e \triangleq X_s - X_{veh} \quad (1)$$

yielded the following state-space description for sensation error e . In addition to the sensation error, the washout filter has to take the simulator state X_d into account, which is required to guarantee that the simulator obeys its constraints. Defining the simulator state X_d ,

$$X_d = \left[\iint a_x^s dt^2 \quad \iint a_y^s dt^2 \quad \iint a_z^s dt^2 \quad \int \dot{\theta}^s dt \quad \int \dot{\phi}^s dt \quad \int \dot{\psi}^s dt \right]^T \quad (2)$$

where $a_x^s, a_y^s, a_z^s, \dot{\theta}^s, \dot{\phi}^s$ and $\dot{\psi}^s$ are translational accelerations (longitudinal, lateral and vertical) and angular velocities (pitch, roll and yaw) in the simulator platform, respectively. After having the dynamic models for sensation error and simulator state, the problem of designing an optimal washout filter gets form as in (3)

$$J = \int_{t_0}^{t_1} (e^T Q e + u_s^T R u_s + x_d^T R_d x_d) dt \quad (3)$$

where $[t_0, t_1]$ is the simulator drive duration and Q , R and R_d are the weighting matrices. The solution of that optimization problem can be solved by a Riccati equation with MATLAB [9, 16, 22], which gives the transfer matrix $W(s)$ as a result. First, the weighting matrices Q , R and R_d are determined, and afterwards, the corresponding filter $W(s)$ is obtained, which generates u_s from u_{veh} . The procedure is iterated until u_s satisfies all the simulator constraints:

$$\left| \dot{\theta}^s(t) \right| \leq \dot{\theta}_{\max}^s, \quad |a_x^s(t)| \leq a_{x,\max}^s, \quad |\theta^s(t)| \leq \theta_{\max}^s, \quad |V_x^s(t)| \leq V_{x,\max}^s \text{ and } |X^s(t)| \leq X_{\max}^s$$

$$\left| \dot{\phi}^s(t) \right| \leq \dot{\phi}_{\max}^s, \quad |a_y^s(t)| \leq a_{y,\max}^s, \quad |\phi^s(t)| \leq \phi_{\max}^s, \quad |V_y^s(t)| \leq V_{y,\max}^s \text{ and } |Y^s(t)| \leq Y_{\max}^s$$

$$\left| \dot{\psi}^s(t) \right| \leq \dot{\psi}_{\max}^s, \quad |a_z^s(t)| \leq a_{z,\max}^s, \quad |\psi^s(t)| \leq \psi_{\max}^s, \quad |V_z^s(t)| \leq V_{z,\max}^s \text{ and } |Z^s(t)| \leq Z_{\max}^s$$

for $t \in [t_0, t_1]$, where

$$\dot{\theta}_{\max}^s, a_{x,\max}^s, \theta_{\max}^s, V_{x,\max}^s, X_{\max}^s$$

$$\dot{\phi}_{\max}^s, a_{y,\max}^s, \phi_{\max}^s, V_{y,\max}^s, Y_{\max}^s$$

$$\dot{\psi}_{\max}^s, a_{z,\max}^s, \psi_{\max}^s, V_{z,\max}^s, Z_{\max}^s$$

are the maximum allowable simulator platform angular rates, specific forces, velocities and displacements of the simulator platform, respectively.

3. DISCRETE-TIME LQR-BASED MOTION CUEING

The DT LQR control is assured to produce a feedback that stabilizes the system as long as the following basic theorems are fulfilled [23, 24]:

The system (A,B) must be reachable. R and Q must be positive definite. Then, the closed-loop system (A-BK) has to be asymptotically stable. It should be noted that this retains in any case the stability of the open-loop system. The reachability may be verified by checking that the controllability matrix $U = [B \ AB \ A^2B \ \dots \ A^{n-1}B]$ has full rank n .

The system (A,B) must be stabilizable. R must be positive definite, Q must be positive semi-definite and (A, \sqrt{Q}) must be observable. Then, the closed-loop system (A-BK) has to be asymptotically stable [23, 24].

If the tracking problems have a priority, a better performance washout filter must be designed, which utilizes the simulator capability and consequently reduces the sensation error to lower level than that of classical motion cueing [7–9, 15, 16].

When a trajectory is taken as a priority, it is generally described in discrete time [16].

For most cases, it is usually quite challenging that dynamic programming must be solved numerically. A few cases can be solved analytically; discrete LQR is one of them [16, 22]. Thus, DT LQR-based motion cueing was chosen to reduce the sensation error in this paper.

First, sampling u_s at the sampling period $t_s = \frac{(t_1-t_0)}{N}$ gives $u_s(k) = u_s(t_0 + kt_s)$ for $k = 0, 1, \dots, N$. In terms of discretization, the optimization problem becomes the following:

$$J = \sum_{k=0}^N \tilde{e}^T(k) \tilde{Q} \tilde{e}(k) + \tilde{u}_s^T(k) \tilde{R} \tilde{u}_s(k) \quad (4)$$

$$|\dot{\theta}^s(k)| \leq \dot{\theta}_{\max}^s, |a_x^s(k)| \leq a_{x,\max}^s, |\theta^s(k)| \leq \theta_{\max}^s, |V_x^s(k)| \leq V_{x,\max}^s \text{ and } |X^s(k)| \leq X_{\max}^s$$

$$|\dot{\phi}^s(k)| \leq \dot{\phi}_{\max}^s, |a_y^s(k)| \leq a_{y,\max}^s, |\phi^s(k)| \leq \phi_{\max}^s, |V_y^s(k)| \leq V_{y,\max}^s \text{ and } |Y^s(k)| \leq Y_{\max}^s$$

$$|\dot{\psi}^s(k)| \leq \dot{\psi}_{\max}^s, |a_z^s(k)| \leq a_{z,\max}^s, |\psi^s(k)| \leq \psi_{\max}^s, |V_z^s(k)| \leq V_{z,\max}^s \text{ and } |Z^s(k)| \leq Z_{\max}^s$$

The main goal is to convert (4) into a standard quadratic programming (QP) [16] by piling $u_{\text{veh}}(k)$, $u_s(k)$ and $e(k)$ from the 0^{th} sample to the N^{th} sample to obtain \tilde{e} , \tilde{u}_{veh} and $\tilde{u}_s (\in \mathbb{R}^{2(N+1)})$, where \sim is used to denote a pile of vectors. Solving the state equation for sensational error in discrete time gives an expression for \tilde{e} :

$$\tilde{e} = \tilde{K}(\tilde{u}_s - \tilde{u}_{\text{veh}}) \quad (5)$$

is the vestibular model's sensation error [16]. Equation (5) shows a state-variable feedback (SVFB) control for the motion cueing. The cost function J can be consequently represented in a quadratic form:

$$J = \tilde{e}^T \tilde{Q} \tilde{e} + \tilde{u}_s^T \tilde{R} \tilde{u}_s \quad (6)$$

A discrete-time system can be given as $\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\tilde{u}_k$. The aim is to seek an SVFB control $u_k = Kx_k$ that minimizes the discrete-time performance index for the proposed DT LQR motion cueing [23].

$$J(e_k) = \frac{1}{2} \sum_{i=k}^{\infty} (\tilde{e}_i^T \tilde{Q}_e \tilde{e}_i + \tilde{u}_{si}^T \tilde{R}_{si} \tilde{u}_{si}) \quad (7)$$

with design weighting matrices $\tilde{Q}_e = \tilde{Q}_e^T \geq 0$, $\tilde{R} = \tilde{R}^T > 0$. It should be noted that this cost function also depends on all the future control inputs e_k, e_{k+1}, \dots

This is known as the DT LQR problem, because the system is linear and the cost is quadratic [23].

Substituting the SVFB control into this yields

$$J(e_k) = \frac{1}{2} \sum_{i=k}^{\infty} \tilde{e}_i^T (\tilde{Q}_e + \tilde{K}^T \tilde{R} \tilde{K}) \tilde{e}_i \quad (8)$$

The closed-loop system with the closed-loop plant matrix \tilde{A}_c using SVFB becomes

$$\tilde{e}_{k+1} = (\tilde{A} - \tilde{B} \tilde{K}) \tilde{e}_k = \tilde{A}_c \tilde{e}_k \quad (9)$$

A difference equation equivalent to (7) is

$$J(e_k) = \frac{1}{2} (\tilde{e}_k^T \tilde{Q}_e \tilde{e}_k + \tilde{u}_s^T(k) \tilde{R}_s \tilde{u}_s(k)) + \frac{1}{2} \sum_{i=k+1}^{\infty} (\tilde{e}_i^T \tilde{Q}_e \tilde{e}_i + \tilde{u}_{si}^T \tilde{R}_{si} \tilde{u}_{si}) \quad (10)$$

$$J(e_k) = \frac{1}{2} (\tilde{e}_k^T \tilde{Q}_e \tilde{e}_k + \tilde{u}_s^T(k) \tilde{R}_s \tilde{u}_s(k)) + J(\tilde{e}_{k+1}) \quad (11)$$

where one requires the boundary condition $J(e_k = 0) = 0$. That is, if one can solve (11) given a control input sequence, it is the same as finding $J(e_k)$ for the given current state \tilde{e}_k by evaluating the infinite sum in (7) [23, 24].

The optimal cost, minimizing the cost function, is given for all k in the form of (12).

$$J^*(e_k) = \tilde{e}_k^T \tilde{P} \tilde{e}_k \quad (12)$$

That is, suppose the optimal cost is quadratic in terms of the current state and in terms of some unknown kernel matrix P . If we can find the optimal feedback in terms of this assumption, then the assumption will be valid [23].

It should be noted that $J^*(e_k)$ only depends on the initial state \tilde{e}_k , not the future inputs, because the optimal cost is defined by selecting all future feedback controls as $\tilde{u}_{si} = \tilde{K} \tilde{e}_k$ with K the optimal SVFB gain [23, 24].

To find the optimal SVFB and the optimal cost kernel P , we substitute (12) into (11) to obtain

$$\tilde{e}_k^T \tilde{P} \tilde{e}_k = \frac{1}{2} (\tilde{e}_k^T \tilde{Q}_e \tilde{e}_k + \tilde{u}_s^T(k) \tilde{R}_s \tilde{u}_s(k)) + \tilde{e}_{k+1}^T \tilde{P} \tilde{e}_{k+1} \quad (13)$$

Note that both $J(e_k)$ and $J(e_{k+1})$ are expressed in terms of the same kernel P , according to (12). By substituting $\tilde{e}_{k+1} = \tilde{A} \tilde{e}_k + \tilde{B} \tilde{u}_s(k)$ into (12),

$$J(e_k) = \tilde{e}_k^T \tilde{P} \tilde{e}_k = \frac{1}{2} (\tilde{e}_k^T \tilde{Q}_e \tilde{e}_k + \tilde{u}_s^T(k) \tilde{R}_s \tilde{u}_s(k)) + \tilde{A} \tilde{e}_k + \tilde{B} \tilde{u}_s(k)^T \tilde{P} (\tilde{A} \tilde{e}_k + \tilde{B} \tilde{u}_s(k)) \quad (14)$$

The optimal control problem is now to minimize this with respect to $\tilde{u}_s(k)$. To do this, differentiate to obtain

$$0 = \frac{\partial}{\partial \tilde{u}_s(k)} J(\tilde{e}_k) = \tilde{e}_k^T \tilde{P} \tilde{e}_k + \tilde{R}_s \tilde{u}_s(k) + \tilde{B}^T \tilde{P} (\tilde{A} \tilde{e}_k + \tilde{B} \tilde{u}_s(k)) \quad (15)$$

This yields the optimal control as

$$(\tilde{R} + \tilde{B}^T \tilde{P} \tilde{B}) \tilde{u}_s(k) = -\tilde{B}^T \tilde{P} \tilde{A} \tilde{e}_k \quad (16)$$

or

$$\tilde{u}_s(k) = -(\tilde{R} + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} \tilde{e}_k \quad (17)$$

This defines the optimal gain as

$$\tilde{K} = (\tilde{R} + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} \quad (18)$$

To find P, $\tilde{u}_s(k) = -\tilde{K} \tilde{e}_k$ should be substituted into (14) to obtain

$$\tilde{e}_k^T [(\tilde{A} - \tilde{B} \tilde{K})^T \tilde{P} (\tilde{A} - \tilde{B} \tilde{K}) - \tilde{P} + \tilde{Q} + \tilde{K}^T \tilde{R} \tilde{K}] \tilde{e}_k = 0 \quad (19)$$

Because this must hold for all current states e_k , one has the matrix equation

$$(\tilde{A} - \tilde{B} \tilde{K})^T \tilde{P} (\tilde{A} - \tilde{B} \tilde{K}) - \tilde{P} + \tilde{Q} + \tilde{K}^T \tilde{R} \tilde{K} = 0 \quad (20)$$

It has to be noted that this is a Lyapunov equation in terms of the closed-loop system matrix

$$\tilde{A}_c = \tilde{A} - \tilde{B} \tilde{K} \quad (21)$$

Now putting (18) into (21) yields

$$\begin{aligned} & \left[\tilde{A} - \tilde{B}(\tilde{R} + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} \right]^T \tilde{P} \left[\tilde{A} - \tilde{B}(\tilde{R} + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} \right] \\ & - \tilde{P} + \tilde{Q} + \left[(\tilde{R} + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} \right]^T \tilde{R} \left[(\tilde{R} + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} \right] = 0 \end{aligned} \quad (22)$$

after some simplification and determination gives

$$\tilde{A}^T \tilde{P} \tilde{A} - \tilde{P} + \tilde{Q} - \tilde{A}^T \tilde{P} \tilde{B} (\tilde{R} + \tilde{B}^T \tilde{P} \tilde{B})^{-1} \tilde{B}^T \tilde{P} \tilde{A} = 0 \quad (23)$$

This equation is commonly used in modern control theory. It is known as the discrete-time algebraic Riccati equation [23]. At this point, solving the DT LQR problem was succeeded; therefore, all our assumptions were justified [23].

There are some numerical procedures available for solving the discrete-time algebraic Riccati equation. The MATLAB routine, which performs this, is named `dlqr` (A,B,Q,R) [22, 25].

Finally, the DT LQR-based motion cueing algorithm took the form as in (24) after considering the closed-loop system matrix (HP stands for high-pass filtered motion, and LP for low-pass filtered motion), which was discretized with a sampling interval of $T = 1/60$ s to obtain the discrete-time motion cueing algorithms implemented in our dll plugin.

$$\begin{aligned} \begin{bmatrix} x(k+2) \\ y(k+2) \\ z(k+2) \end{bmatrix} &= \begin{bmatrix} \frac{2x(k+1)_{HP}+4.9985}{x(k+2)_{HP}+1} & 0 & 0 \\ 0 & \frac{2y(k+1)_{HP}+4.9985}{y(k+2)_{HP}+1} & 0 \\ 0 & 0 & \frac{2z(k+1)_{HP}+4.9985}{z(k+2)_{HP}+1} \end{bmatrix} \cdot \begin{bmatrix} x(k+1) \\ y(k+1) \\ z(k+1) \end{bmatrix} \\ &+ \begin{bmatrix} -\arcsin\left(\frac{x(k+2)_{LP}+509.19}{9.81}\right) & 0 & 0 \\ 0 & -\arcsin\left(\frac{y(k+2)_{LP}+509.19}{9.81}\right) & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta(k) \\ \phi(k) \\ \psi(k) \end{bmatrix} \\ &- \begin{bmatrix} 41 & 0 & 0 \\ 0 & 41 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta(k+1) \\ \phi(k+1) \\ \psi(k+1) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{10x(k)_{HP}-3.4543}{x(k+2)_{HP}+1} & 0 & 0 \\ 0 & \frac{10y(k)_{HP}-3.4543}{y(k+2)_{HP}+1} & 0 \\ 0 & 0 & \frac{10z(k)_{HP}-3.4543}{z(k+2)_{HP}+1} \end{bmatrix} \cdot \begin{bmatrix} x(k) \\ y(k) \\ z(k) \end{bmatrix} \end{aligned} \quad (24)$$

3.1. Fast Fourier transform

An FFT is an algorithm to compute the discrete Fourier transform (DFT) and its inverse (25 and (26)) [26].

For this study, the roll acceleration temporal runs were converted into frequency domain by FFT, which is allowing us to achieve frequential analysis.

$$X(k) = \sum_{j=1}^N x(j) \omega_N^{(j-1)(k-1)} \quad (25)$$

From (25), (26) can be obtained as its inverse transform [26]:

$$x(j) = \frac{1}{N} \sum_{k=1}^N X(k) \omega_N^{-(j-1)(k-1)} \quad (26)$$

where $\omega_N = e^{\frac{-2\pi i}{N}}$ is the N^{th} root of unity [26].

MATLAB uses the functions $Y = \text{fft}(x)$ and $y = \text{ifft}(X)$ to implement the FFT and inverse transform pair given for vectors of length N by (25) and (26) [26].

3.2. Data denoising

The wavelet transform is used as a mathematical instrument in order to analyze any non-stationary time series, showing the temporal variability of the PSD [27].

The data collected from SCANNERSTUDIO driving simulation software were denoised by using Daubechies wavelets (dbN) 1D wavelet packet of MATLAB. For the entropy, Shannon type [28–30] was chosen for the research pursued in this paper.

In dbN , N is the order. More detailed information about Daubechies wavelets can be found in [30, 31]. By typing `wavemenu` at the MATLAB command prompt, the GUI of the wavelet can be called. After the GUI comes to the apparent window, both the cabin and motion platform level roll accelerations of the two washout algorithms (classical and DT LQR algorithms) are loaded separately and independently to the 1D wavelet packet to denoise (Figure 3). `db4` was used as 1D wavelet packet of MATLAB to process the noisy signals. For entropy, Shannon type was assigned [30].

Soft thresholding method [32] as fixed form threshold (unscaled wn) was applied to process the noisy roll acceleration signals from both cabin and platform levels in the wavelet 1D denoising window. The global threshold was adjusted to 80% of the maximum global threshold. Lastly, by pressing the ‘De-noise button’, the denoised signal was obtained.

4. RESULTS AND DISCUSSION

In this article, two different motion cueing algorithms were proposed: classical and optimal (DT LQR) algorithms. Their roll-jerk-based sensation errors (Tables III and IV) were compared in order to determine the sensory cue conflict [33, 34] conditions for the same scenario chosen (Figure 5). Also, the natural frequencies and their corresponding amplitudes were discussed.

The experiments were driven on two conditions of motion cueing: classical and optimal (DT LQR) algorithms (Figure 4). According to the proposed approach, the sensation error was aimed to decrease in order to decrease the simulator sickness with the implementation of the optimal (DT LQR) algorithm on the simulator compared with the already implemented classical algorithm.

Figure 1 demonstrates the block diagrams of the used DT LQR and classical motion cueing algorithms. The dashed lines indicates the strategy for the DT LQR, whereas the continuous lines show the classical strategy. It can easily be identified that the classical strategy is an open loop; inversely, the DT LQR one is a closed-loop system with an SVFB that enables to control the dynamics of the platform.

Figure 5 illustrates a specific country road scenario that was used in the experimental stage. The red curve shows the trajectory followed by the vehicle.

1-D Wavelet Packet Denoising

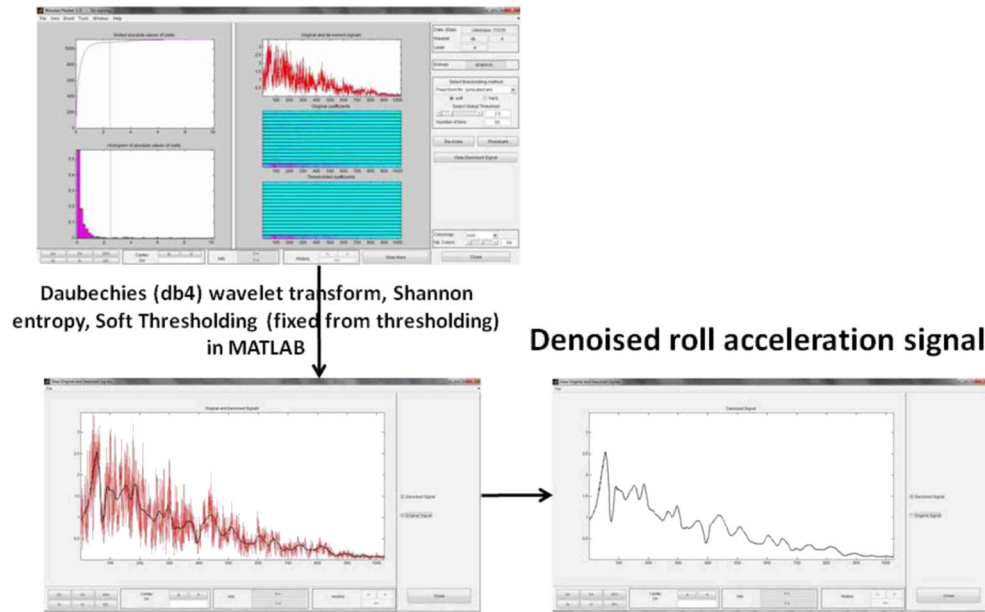


Figure 3. Denoising procedure.

Motion Cueing Algorithms

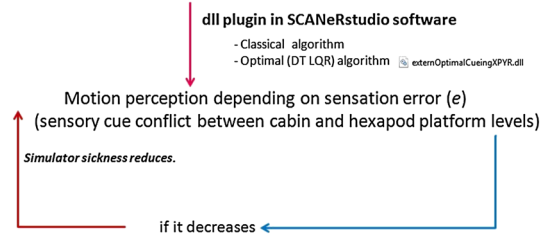


Figure 4. Motion cueing algorithms integration in the simulator.



Figure 5. Trajectory of the scenario driven in the experiments.

According to the scenario, the vehicle goes straight, takes the first left turn, continues till the right turn is reached, takes the right turn and drives till the crossroad section of the scenario. Then, it stops with a full brake (Figure 6).

During the straight direction, the vehicle drives at a constant speed of 60 km/h for 126 s (Figure 6).

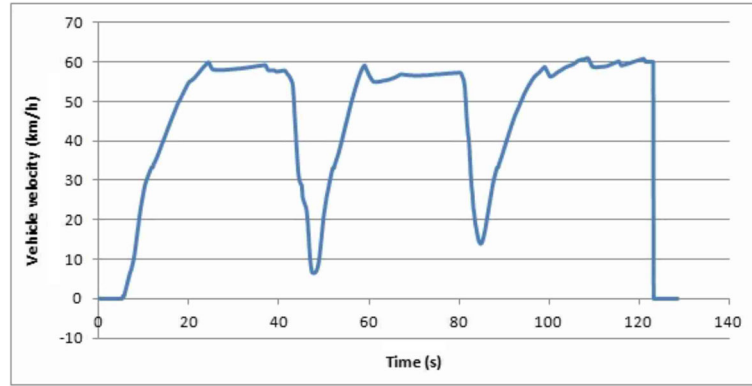


Figure 6. Vehicle velocity profile for simulator experiments.

Table I. SAAM hexapod platform capability.

DOF	Displacement	Velocity	Acceleration
Pitch	$\pm 22^\circ$	$\pm 30^\circ/\text{s}$	$\pm 500^\circ/\text{s}^2$
Roll	$\pm 21^\circ$	$\pm 30^\circ/\text{s}$	$\pm 500^\circ/\text{s}^2$
Yaw	$\pm 22^\circ$	$\pm 40^\circ/\text{s}$	$\pm 400^\circ/\text{s}^2$
Heave	$\pm 0.18 \text{ m}$	$\pm 0.3 \text{ m/s}$	$\pm 0.5 \text{ g}$
Surge	$\pm 0.25 \text{ m}$	$\pm 0.5 \text{ m/s}$	$\pm 0.6 \text{ g}$
Sway	$\pm 0.25 \text{ m}$	$\pm 0.5 \text{ m/s}$	$\pm 0.6 \text{ g}$

Table I shows the maximum limits of translational and angular displacements, velocities and accelerations of the hexapod platform for the SAAM driving simulator [7–9] where gravitational acceleration is given by $g \approx 9.81 \text{ m/s}^2$.

The implemented washout algorithms took the constraints of the hexapod platform of the simulator into account (Figure 1; Table I).

Table II summarizes the motion cueing parameters used for the dynamics operation of SAAM hexapod platform automobile simulator [7–9].

Those parameters were applied for both algorithms. As an exception, the DT LQR strategy had an SVFB.

Figure 7 resumes the basic logic of dynamic automobile simulators. This paper discussed the motion drive algorithms' effects on simulator sickness by integrating two different kinds of motion cueing strategy.

The DT LQR algorithm was developed to have a less sensation error (for the platform to cabin level change, refer to Tables III and IV); thus, the simulator sickness could be decreased.

The sensory-conflict-based sickness was evaluated at the vehicle–platform levels in frequency domain (Figures 8 and 9). The simple curves demonstrate the classical motion cueing algorithm, whereas the dashed ones illustrate the optimal (DT LQR) algorithm in Figures 8 and 9.

Table II. Motion cueing parameters for the dynamic operation for a 6-DOF hexapod platform.

Classical motion cueing	Longitudinal (gain = 1)	Lateral (gain = 1)	Roll (gain = 0.25)	Pitch (gain = 0.25)	Yaw (gain = 1)
Second-order LP cut-off frequency (Hz)			0.3	0.3	
Second-order LP damping factor			0.7	0.7	
First-order LP time constant (s)	0.1	0.1			0.1
Second-order HP cut-off frequency (Hz)	0.5	0.5			2
Second-order HP damping factor	1	1			1
First-order HP time constant (s)	2	2			2

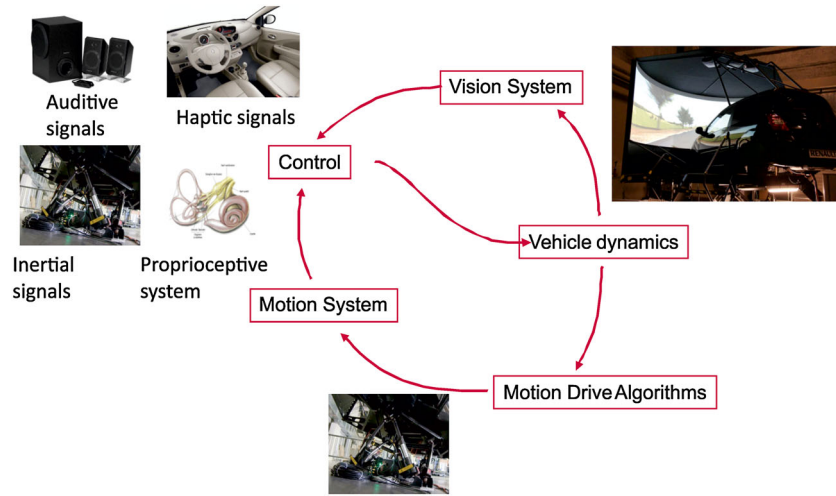


Figure 7. Multi-sensory interaction in dynamic driving simulator.

Table III. Cabin roll acceleration based results in frequency domain.

	Classical algorithm	Optimal (DT LQR) algorithm
Natural frequencies (Hz)	1.8–2.2	1.8–2.5
Corresponding amplitudes to natural frequencies ($^{\circ}/s^2$)	0.5–0.34	0.62–0.48
Total sensed roll jerk ($^{\circ}/s^3$)	0.92	1.30

Table IV. Platform roll acceleration based results in frequency domain.

	Classical algorithm	Optimal (DT LQR) algorithm
Natural frequencies (Hz)	0.4–1.5	1.5–2.2
Corresponding amplitudes to natural frequencies ($^{\circ}/s^2$)	2.6–1.7	1.7–1.8
Total sensed roll jerk ($^{\circ}/s^3$)	5.76	4.05

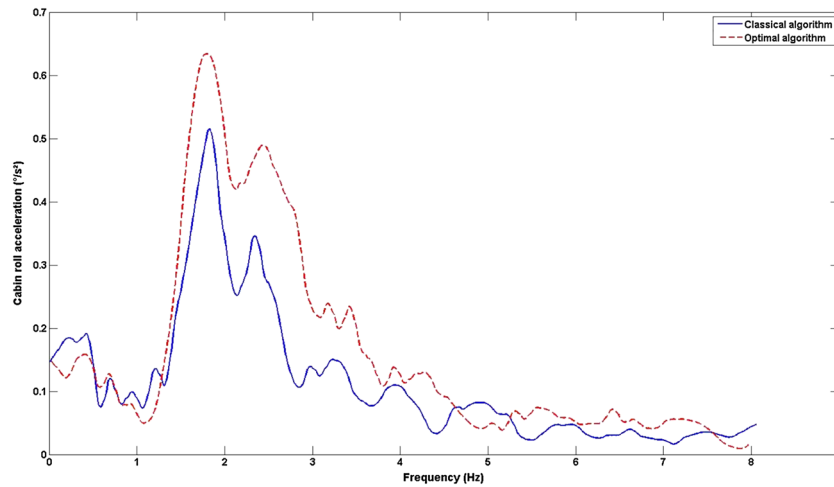


Figure 8. Cabin roll acceleration frequential analysis.

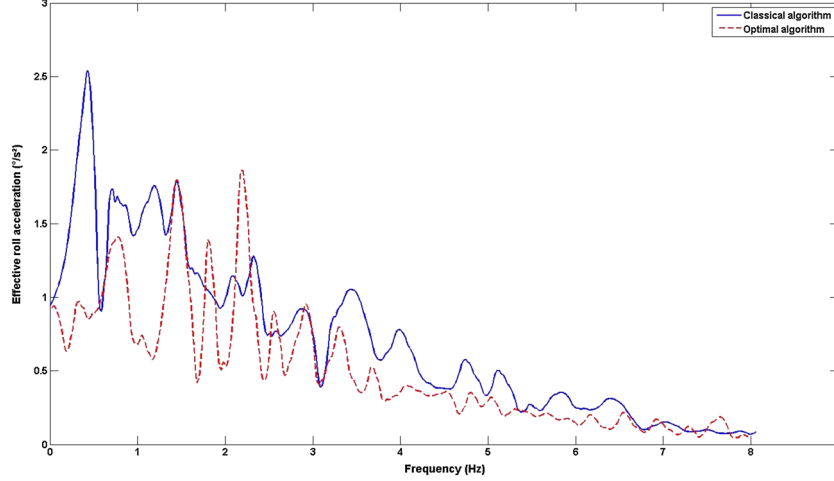


Figure 9. Platform roll acceleration frequential analysis.

Equation (27) calculates the total sensed roll jerk at cabin (vehicle) level. This level signifies the vehicle dynamics model that moves in the visual environment during the simulator drive experiments. The vehicle model has to be created depending on the data (kinematics data: camber angle, toe-in and toe-out, etc; dynamics data: suspension, damper, tires, etc.) of a real vehicle in the driving simulation software [35]. It should be denoted that same passenger car model was used for both algorithms during the experimental phase.

$$\text{Cabin level total sensed roll jerk} = \int_0^f \ddot{\phi}_{\text{veh}} \cdot df \quad (27)$$

where $\ddot{\phi}_{\text{veh}}$ is the roll acceleration of the vehicle (cabin) in $^\circ/\text{s}^2$ and f is the frequency in Hz.

Equation (28) calculates the total sensed roll jerk at simulator platform level. Jerk is defined as the first derivative of the acceleration. This level signifies the hexapod platform dynamics that moves inertially during the simulator drive experiments (Figure 7).

Depending on the motion cueing type whether it is an open-loop control or a closed-loop control (Figures 1 and 4), the platform dynamics response changes (Figure 9).

$$\text{Simulator platform level total sensed roll jerk} = \int_0^f \ddot{\phi}_s \cdot df \quad (28)$$

where $\ddot{\phi}_s$ is the roll acceleration of the simulator platform in $^\circ/\text{s}^2$ and f is the frequency in Hz.

In the postprocessing phase of the real-time data registered during the simulator drives, ‘trapz’ command was used in MATLAB to compute the total sensed roll jerk at both algorithms for cabin–platform levels.

Resonant frequency in roll was declared as in the range of 2 Hz or more in vehicle body (sprung mass) [36]. Realistic natural roll frequency values were explained as 1.5 Hz for a passenger car [37]. From another research, the findings showed that lateral motion (roll, yaw, etc.) in the frequency range of 0.1–0.5 Hz induces car sickness [38].

Tables III and IV show the frequential analysis results of the measured roll accelerations for both vehicle and platform in real-time simulator experiments.

It is seen from these tables that the roll jerk sensitivities ($5.76/0.92 \approx 6.26$ for classical and $4.05/1.30 \approx 3.12$ for DT LQR algorithm, finally $6.26/3.12 \approx 2$) from platform to cabin stages decreased approximately two times by using optimal (DT LQR) strategy.

5. CONCLUSION AND FUTURE WORK

This study investigated the effect of a closed-loop tracking control (DT LQR-based optimal control strategy) of the hexapod platform on sensation error (between vehicle and platform levels).

Having optimal (DT LQR) algorithm provided two times less contradicting cues between visual (vehicle) and inertial (platform) signals for the same scenario driven with the identical conditions.

Regarding the motion sickness, classical motion cueing presented a sickness incidence because the motion platform yielded a natural roll frequency of 0.4 Hz (Table IV). According to [38], it is provocative to induce sickness. On the other hand, using DT LQR algorithm increased the natural roll frequency to 1.5 Hz for the sprung mass (vehicle body) that is convenient for a passenger car [37].

For prospective research, the hybrid (discrete-time model reference adaptive control + DT LQR) motion cueing performance will be assessed in multi-sensory level (vehicle, platform) and their interaction with biofeedback response methods (EMG and postural stability) for the dynamic simulator.

ACKNOWLEDGEMENTS

Arts et Metiers ParisTech built up the SAAM driving simulator with the partnership of Renault and Grand Chalon. This research was realized in the framework of the geDRIVER project.

REFERENCES

1. Angelaki DE, Gu Y, DeAngelis GC. Multisensory integration: psychophysics, neurophysiology and computation. *Current Opinion in Neurobiology* 2009; **19**(4):452. NIH Public Access.
2. Kolasinski EM. Simulator sickness in virtual environments, 1995. DTIC Document.
3. Curry R, Artz B, Cathey L, Grant P, Greenberg J. Kennedy ssq results: fixed-vs motion-based FORD simulators. *Proceedings of DSC*, Paris, France, 2002; 289–300.
4. Watson G. A synthesis of simulator sickness studies conducted in a high-fidelity driving simulator. *Proceedings of Driving Simulation Conference*, Paris, France, 2000; 69–78.
5. Stewart D. A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers* 1965; **180**(1):371–386. SAGE Publications.
6. Meywerk M, Aykent B, Tomaske W. Einfluss der Fahrdynamikregelung auf die Sicherheit von N1-Fahrzeugen bei unterschiedlichen Beladungszuständen. Das elektronische BAST-Archiv: Elba, 2009. Teil 2: Fahrversuche und Fahrsimulatorversuche. FE 82.329/2007.
7. Aykent B, Paillot D, Mérienne F, Fang Z, Kemeny A. *Study of the Influence of Different Washout Algorithms on Simulator Sickness for a Driving Simulation Task*. ASME: Venice, Italy, 2011. ISBN: 978-0-7918-4432-8, pp. 331–341.
8. Aykent B, Paillot D, Merienne F, Kemeny A. The influence of the feedback control of the hexapod platform of the SAAM dynamic driving simulator on neuromuscular dynamics of the drivers. *Driving Simulation Conference (DSC) 6-7 September 2012*, Paris, France, 2012; 1–2.
9. Aykent B, Paillot D, Merienne F, Kemeny A. A LQR washout algorithm for a driving simulator equipped with a hexapod platform: the relationship of neuromuscular dynamics with the sensed illness rating. *CONFERE, 5-6 July 2012*, Venice, Italy, 2012; 1–9.
10. Kim MS, Moon YG, Kim GD, Lee MC. Partial range scaling method based washout algorithm for a vehicle driving simulator and its evaluation. *International Journal of Automotive Technology* 2010; **11**(2):269–275. Springer.
11. Reymond G, Kemeny A. Motion cueing in the Renault driving simulator. *Vehicle System Dynamics* 2000; **34**(4):249–259. Taylor & Francis.
12. Kemeny A, Panerai F. Evaluating perception in driving simulation experiments. *Trends in Cognitive Sciences* 2003; **7**(1):31–37.
13. Chen D, Hart J, Vertegaal R. Towards a physiological model of user interruptability. *Human-Computer Interaction-INTERACT 2007*, Springer, 2007; 439–451.
14. Pick AJ, Cole DJ. Dynamic properties of a driver's arms holding a steering wheel. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 2007; **221**(12):1475–1486.
15. Levison WH, Junker AM. A model for the pilot's use of motion cues in steady-state roll-axis tracking tasks. *Flight Simulation Technologies Conference*, Arlington, Texas, 1978; 149–159.
16. Cho YM, Kim IK, Kam MS, Kim HS, Hur SM, Min KD. Fast design of the QP-based optimal trajectory for the Eclipse-11 motion simulator. *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, Vol. 2, Maui, HI, 2003; 1232–1237.

17. Sivan R, Ish-Shalom J, Huang JK. An optimal control approach to the design of moving flight simulators. *IEEE Transactions on Systems, Man and Cybernetics* 1982; **12**(6):818–827.
18. Cardullo F, Telban R, Houck J. Motion cueing algorithms: a human centered approach. *5th International Symposium on Aeronautical Sciences*, Zhukovsky, Russia, 1999; 1–11.
19. Telban RJ, Cardullo FM, Houck JA. A nonlinear, human-centered approach to motion cueing with a neurocomputing solver. *Technical Report*, NASA, 2002. AIAA 2002-4692, pp. 1–10.
20. Telban RJ, Cardullo FM, Guo L. Investigation of mathematical models of otolith organs for human centered motion cueing algorithms. *Proceedings of the AIAA Modelling and Simulation Technologies Conference*, Vol. 17, Denver (CO), 14 August 2000; 1.
21. Scokaert POM, Rawlings JB. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control* 1998; **43**(8):1163–1169.
22. Hanselman DC, Littlefield B. *Mastering Matlab 7*. Pearson Prentice Hall: New Jersey, USA, 2005.
23. Lewis FL. *Feedback Control for Discrete-Time Systems*, Lecture Notes. Advanced Controls and Sensors Group, The University of Texas Arlington: Arlington, TX, USA, 2008. Updated: Wednesday, November 05, 2008.
24. Xu X. Discrete time mixed LQR/ H_∞ control problems, 2011.
25. The Mathworks Inc. Control system toolbox for use with MATLAB, 1998. Computation Visualization Programming. User's Guide Version 4.1.
26. The Mathworks Inc. MATLAB 7 function reference: A - E, 2008.
27. Bolzan MJA, Guarnieri FL, Vieira PC. Comparisons between two wavelet functions in extracting coherent structures from solar wind time series. *Brazilian Journal of Physics* 2009; **39**(1):12–17. SciELO Brasil.
28. Coifman RR, Wickerhauser MV. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory* 1992; **38**(2):713–718.
29. Donoho DL, Johnstone IM. Ideal denoising in an orthonormal basis chosen from a library of bases. *Comptes Rendus de l'Académie des Sciences. Série I, Mathématique* 1994; **319**(12):1317–1322. Paris: Gauthier-Villars, c1984-c2001.
30. Misiti M, Misiti Y, Oppenheim G, Poggi JM. *Matlab User's Guide: Wavelet Toolbox 4*. The Math Works Inc.: MA, USA, 2008.
31. Daubechies I. *Ten Lectures on Wavelets*, Vol. 61. Society for Industrial and Applied Mathematics: Philadelphia, PA, 1992.
32. Donoho DL. De-noising by soft-thresholding. *IEEE Transactions on Information Theory* 1995; **41**(3):613–627.
33. Oman C. Sensory conflict in motion sickness: an observer theory approach, 1989. NASA, Ames Research Center, Spatial Displays and Spatial Instruments 15 p (SEE N 90-22918 16-54).
34. Oman CM. Motion sickness: a synthesis and evaluation of the sensory conflict theory. *Canadian Journal of Physiology and Pharmacology* 1990; **68**(2):294–303. NRC Research Press.
35. Meywerk M, Aykent B, Tomaske W. *Einfluss der Fahrdynamikregelung auf die Sicherheit von NI-Fahrzeugen bei Unterschiedlichen Beladungszuständen*. Das elektronische BAST-Archiv: Elba, 2009. Teil 1: Grundlagen, Unfallstatistik, Abstütz- und Beladungseinrichtung, Fahrzeugdatenermittlung. FE 82.329/2007.
36. Winkler CB, Blower D, Ervin RD. Rollover of heavy commercial vehicles, 1999. No. UMTRI-99-19.
37. Dixon J. *The Shock Absorber Handbook*. Wiley: West Sussex, England, 2007.
38. Cheung B, Nakashima A. A review on the effects of frequency of oscillation on motion sickness, 2006. no. DRDC-TR-2006-229. DTIC Document.