MODEL PREDICTIVE MOTION CUEING ALGORITHM FOR A TRUCK SIMULATOR

E. Thöndel
Department of Electric Drives and Traction
Czech Technical University
166 27 Prague
Czech Republic
E-mail: thondee@fel.cvut.cz

KEYWORDS
Motion Cueing Algorithm, washout algorithm, Model Predictive Control, simulator, truck simulator, motion platform.

ABSTRACT

The paper looks at the possibilities of applying a model predictive control (MPC) algorithm to control and optimize the motion cueing algorithm (MCA) employed in state-of-the-art simulators for the purposes of reproducing motion effects (in particular acceleration). Typically, MCAs feature a large number of parameters which need to be configured with respect to both the kinematic and dynamic limitations of the motion platform used and the human motion perception thresholds. When an MPC algorithm is employed, these constraints can be fully respected and the ultimate results can, at least in a sense, be considered optimal. The paper shows the possibilities of interlinking MPC with MCA to dynamically adjust the applicable constraints. In addition, all theoretical results will be verified by means of simulations. Since the research described in this paper has been conducted as part of an advanced truck simulator project pursued by the author the parameters chosen for all simulations respect the properties of the motion platform employed in this project.

INTRODUCTION

Motion cueing is an essential part of most professional training and research simulators. In 2011, a new project, supported by the Technology Agency of the Czech Republic, was commenced in order to develop a new type of truck simulator intended primarily for emergency situation training and follow-up research (http://www.vyprask.eu). Emergency situations require an especially high level of motion cueing fidelity, as drivers often react purely instinctively in such circumstances and motion cues are the first indicator of an emergency situation or erroneous vehicle behaviour, such as a punctured tyre etc. Motion cueing fidelity is therefore the cornerstone of the project.
Figure 1 below shows the truck simulator being developed as part of the project and featuring a hydraulic motion platform with six degrees of freedom (hexapod). Known primarily from flight simulators, this type of mechanical system combines the highest degree of movement flexibility with exceptional robustness and rigidity, thus requiring no stabilising frames. In recent years, hydraulic cylinders would often be replaced with electromechanical actuators (a description of the properties and comparison of both actuator types are provided in (Thöndel 2011)). Nevertheless, a hydraulic system was used in this particular instance, being both cheaper and easier to maintain in operation.

Fig. 1: Truck simulator mounted on a motion platform

The core component of the motion cueing algorithm (MCA) is a washout filter, converting data provided by the mathematical-physical vehicle dynamics model (acceleration) to motion of the platform. Various modified or enhanced versions of the basic washout filter structure have been described in international literature (an overview is provided for instance in (Jamson 2010)); all of them, however, feature a set of parameters which have to be configured with respect to the type of motion platform used to make sure that the kinematic and dynamic limits are observed.
The setting-up of these parameters is a rather lengthy iterative process, with the programmers often having to rely on their experience and intuition. Therefore, potential ways of setting up the parameters and objective methods of assessing simulation quality have constantly been hot topics in expert circles.
This paper follows on the author's previous article (see Thöndel 2012) looking at the possibilities of partially automating the configuration process. In this paper the original washout filter is enhanced with model predictive control (MPC) features, which, in essence, directly respect

the platform's kinematic limits as well as the sensory thresholds of the human vestibular system.

MPC AND WASHOUT ALGORITHM

Basic Concept of MPC Algorithm

The fundamental principle of an MPC algorithm is to find the optimum control sequence for a dynamic system described by the state model

$$\begin{aligned}\dot{X} &= AX + BU \\ Y &= CX + DU\end{aligned} \quad (1)$$

where A is the state matrix, B the input matrix, C the output matrix, D the input-to-output matrix, X the state variable, U the control variable and Y the system output.
In addition, the control sequence has to respect the applicable control and state variable constraints.

$$\begin{aligned}U_{min} &\leq U(k) \leq U_{max} \\ X_{min} &\leq X(k) \leq X_{max}\end{aligned} \quad (2)$$

The optimality criterion can be defined with a quadratic criterion function with the weighing matrices Q and R.

$$J = \frac{1}{2}X^T(N)QX(N) + \frac{1}{2}\sum_{k=0}^{N-1}\left(X^T(k)QX(k) + U^T(k)RU(k)\right) \quad (3)$$

However, this paper does not seek to provide a comprehensive description of the MPC algorithm, but rather an outline of its use in MCA. In somewhat simplified terms, the iterative process of the MPC algorithm can be described using the following sequence:

1. determine the current state of the system X
2. calculate the optimum control sequence for the given prediction horizon based on the initial state, an estimate of the reference (desired) value and predicted system state development using the system model
3. apply the first control sequence value on the system.

MPC algorithms provide several key benefits. In particular, they ensure optimum control with respect to the given criterion function and allow a clear definition of control and state variable constraints. On the other hand, MPC algorithms usually require relatively high computing power, as system development forecasts (simulations) for the given prediction horizon are calculated during every algorithm iteration. However, with the computing performance of modern computers rising MPC algorithms have been becoming increasingly popular also in highly dynamic processes.

Motion Cueing Algorithm

In simulation technology, MCA is used to generate motion cues (acceleration). The primary MCA component is a washout filter (WF), transforming the input acceleration value to the position of the motion platform.
The original WF structure is non-linear, being derived from the basic concept of reproducing acceleration by inclining the platform:

$$a = g \sin\varphi \quad (4)$$

where a is reproduced acceleration, g is gravitational acceleration and φ the platform's inclination angle.
After linearizing the above function ($a \cong g\varphi$), the linearized form of the washout algorithm can be described with the following equations of state:

$$A = \begin{bmatrix} [A_h]_{3x3} & [A_m]_{3x3} \\ [0]_{3x3} & [A_l]_{3x3} \end{bmatrix}$$

$$B = \begin{bmatrix} [B_h]_{3x1} \\ 1/g[B_l]_{3x1} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 \quad 0 \quad 1 \quad g \quad 0 \quad 0 \\ [I]_{6x6} \end{bmatrix} \quad (5)$$

$$D = [0]_{7x1}$$

$$X = [x \quad \dot{x} \quad \ddot{x} \quad \varphi \quad \dot{\varphi} \quad \ddot{\varphi}]^T$$

$$Y = \begin{bmatrix} a \\ X \end{bmatrix}$$

where A_h and B_h are filter matrices transforming high-frequency acceleration to linear motion, A_l and B_l are filter matrices transforming low-frequency acceleration to rotational motion and A_m is the mutual relationship between the high-frequency and low-frequency components.
The typical WF has to be set up with respect to the kinematic limits of the motion platform used and sensory thresholds of the human vestibular system. In an ideal scenario, the filters can be set up so as to correspond exactly with the characteristics of the human vestibular system (see, for instance, the transfer model described in (Telban et al. 2000)). In practice, however, a compromise always has to be made between response speed and simulation fidelity. This, in turn, means that correct configuration requires a certain degree of experience on the part of the simulation engineer and feedback from validation experiments.
The rest of this paper presents a different view on MCA, using the benefits brought by modern control theory.

MPC-MCA Algorithm

Once WF has been described with state matrices, control theory results can be applied to optimize WF feedback. In theory, the dynamic behaviour of a linear stationary system can be altered in any way by means of state feedback control, assuming all system states are controllable (Havlena and Štecha 2002).

Def. 1: State X is controllable assuming there is a control U capable of transferring the initial state X to the zero state. The system as a whole is controllable if all states are controllable.

Def. 2: A linear stationary system described by the state matrices

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_n \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (6)$$

is always controllable (Frobenius normal form).

After writing the MCA matrices in the above form (6), the linear MCA system is fully controllable and its dynamic behaviour can thus be, theoretically, altered in any way by means of state feedback control. Clearly, there are obvious limitations in practice, given primarily by the motion platform's kinematic and dynamic limits and human sensory thresholds. As an additional benefit, all states of the above form allow a clear physical interpretation.

Modern control methods are often based on different optimization approaches, with some criteria (such as stability) being fulfilled implicitly. One of these methods, the MPC algorithm has the additional benefit of being able to work with clearly defined state and control variable constraints. This allows a direct integration of the motion platform's kinematic and dynamic limits and human sensory threshold values into the equation as input parameters.
Combining MPC with MCA results in the following feedback control structure:

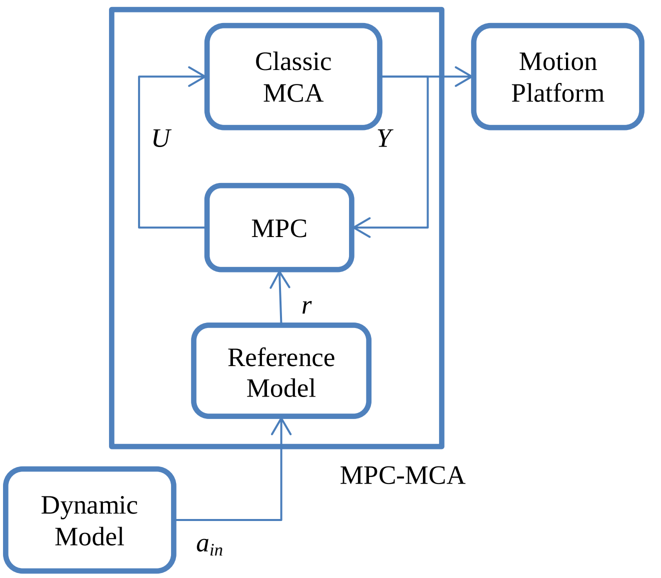

Fig. 2: MPC-MCA algorithm structure

The following parameters have to be defined when configuring the MPC algorithm:

- control and state variable constraints,
- reference model (model forecasting the development of the desired variable), and
- prediction horizon.

Constraints

Essentially, the constraints are the control and state variable limit values to be respected by the MPC algorithm when determining the optimum control sequence for the given prediction horizon.
Most of the constraints are constant, given by human physiology (Table 1), in particular the sensory thresholds of the vestibular system (see, for instance, (Groen and Jongkees 1948)). In other words, the required inclination of the motion platform, as defined in (4), ought, ideally, to be reached below these thresholds to ensure that humans perceive the manoeuvre as accelerating translational motion and not a tilt. In practice, however, a compromise often has to be made between simulation fidelity and feedback speed, resulting in the sensory thresholds being exceeded.

Table 1: Vestibular system sensory thresholds

Angular acceleration	0.3 deg/s^2
Angular velocity	3 deg/s
Translational acceleration	0.1 m/s^2

The second set of constraints is determined by the kinematic limits of the motion platform. The motion platform shown in Fig. 1 is a parallel manipulator with six degrees of freedom, also known as a hexapod or Stewart platform. Because of the parallel arrangement of the system, the maximum displacement in a given direction is not constant but rather determined by the current position of the platform. An analytical description of this phenomenon is highly complicated, given the high degree of complexity in expressing direct kinematic transformation (Tomagoj and Budin 2002).
The standard WF and its enhanced versions scale the required input acceleration value to make sure that the motion platform's position remains within the valid kinematic range even in the worst case scenario. This, however, means that the motion platform is not used to its full potential.
On the other hand, the MPC algorithm continuously predicts the behaviour of the controlled system by means of its mathematical model. In this way, potential conflicts with the platform's kinematic limits can be detected based on this forecast and immediately addressed by adjusting (decreasing) the constraints for the maximum displacement in the affected direction. Hence, the MPC algorithm can be considered a variable scaling factor of the required input acceleration value.
Iterative adjustments of the maximum displacement parameter can be performed for instance with this fast and straightforward loop (interval halving):

1. Execute MPC calculation.
2. Calculate the inverse kinematic transformation for the last position in the prediction horizon.
3. Is the last prediction horizon position valid?
    a. YES: increase the constraint

$$X_{max}(k+1) = \frac{1}{2}(X_{max}(k) + X_M) \quad (7)$$

    b. NO: decrease the constraint

$$X_{max}(k+1) = \frac{1}{2}\left(X_{max}(k) + X(k)\right) \qquad (8)$$

4. Return to step 1.

X_M in equation (8) is the theoretical maximum kinematic limit of the motion platform given by its manufacturer. Table 2 below provides an overview of the kinematic limits of the motion platform used in Fig. 1.

Table 2: Maximum kinematic ranges of the motion platform

Pitch	±16.3°
Roll	±17.5°
Yaw	±17.5°
Longitudinal Motion	±0.198 m
Lateral Motion	±0.220 m
Vertical Motion	±0.200 m

Reference Model

The reference model predicts the development of a desired variable – in this specific scenario vehicle acceleration. In a situation where no data regarding control element state and road properties is available, the easiest solution is using a constant value throughout the whole prediction horizon.

$$r(k+1) = r(k)$$
$$r(0) = a_{in} \qquad (9)$$

This reference model type provides good results and can be implemented very easily (see Fig. 3). Several enhancements of the model are being considered:

- using an AR (autoregressive) model with continuous parameter identification, i.e. the reference signal would be estimated based on its previous values,
- closer integration of the dynamic vehicle model, i.e. the reference signal would be estimated with respect to additional vehicle status data (control element positions, road conditions and shape, etc.)

Nevertheless, a more complex reference model also increases the algorithm's computing power requirements. At the same time, the associated benefits in the form of increased simulation fidelity remain uncertain.

Prediction Horizon

The prediction horizon, i.e. the time frame for which the system state development forecast is calculated, has to be selected with respect to the following aspects:

- calculation complexity – the algorithm is expected to be used in a simulator control system and the calculations have to take place in real time,
- control stability,
- sufficiently early detection of possible conflicts with the platform's kinematic limits to make sure that the event can be addressed adequately and with respect to all remaining restrictions

An empirically established prediction horizon of 2 seconds has been used in each iteration of the following simulations.

SIMULATION RESULTS

Figure 3 shows the simulation results and a comparison to a standard MCA (open control loop pursuant to Fig. 2).

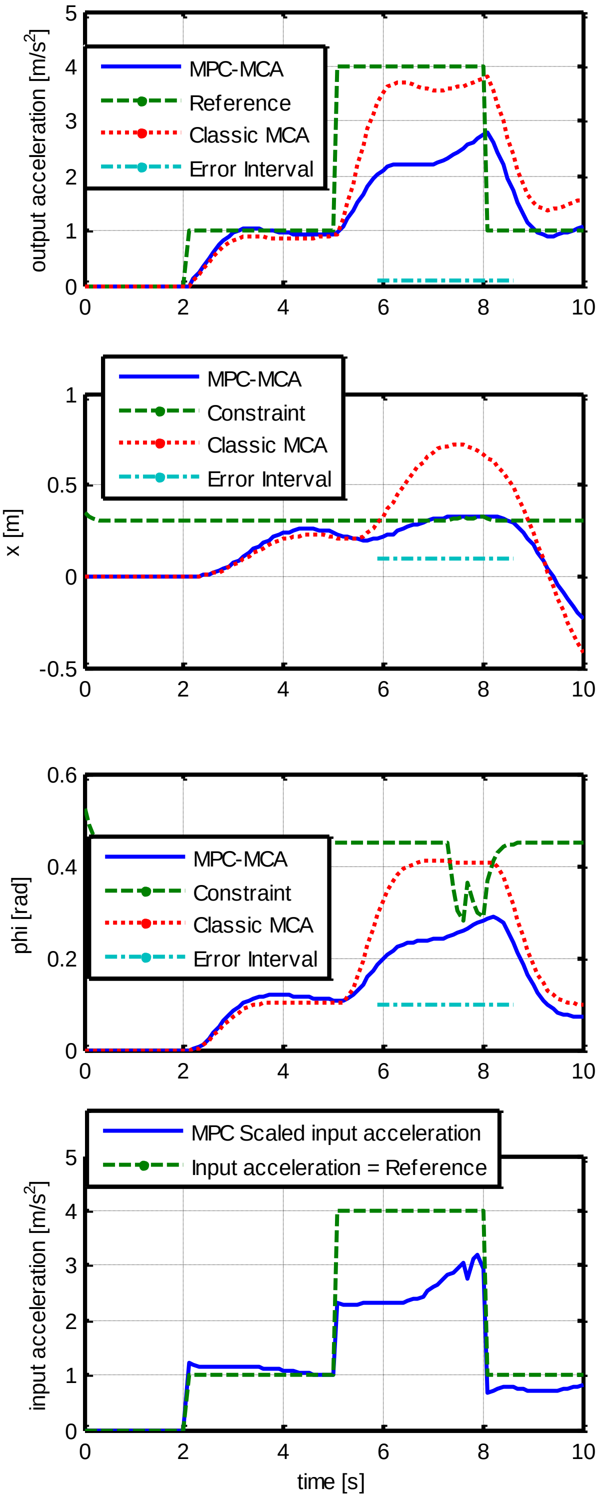

Fig. 3: Simulation results

Standard MCA, a solution not working with the maximum kinematic limits of the motion platform, would, theoretically, reproduce the required acceleration more accurately; however, both platform inclination and displacement limits are exceeded in this case, making the position invalid (the error interval is highlighted in the chart).

MPC-MCA quickly identifies the maximum kinematic limits shortly after the simulation is started and temporarily adjusts (decreases) MPC algorithm constraints as soon as the forecast shows a risk of a kinematic limit being exceeded.

This approach ensures that the full kinematic range of the motion platform can be used, providing, at the same time, the highest acceleration vector reproduction fidelity. In addition, the above solution is optimal from the MPC perspective.

The simulations of the MPC-MCA algorithm described above have been conducted in the Matlab-Simulink environment using the MPC Toolbox (MathWorks 2013).

FOLLOW-UP WORK

The MCA enhancement described in this paper was verified by means of simulations in the MATLAB/Simulink environment. As the next step, the new method will be implemented in the control system of an actual simulator. As the simulator in question is equipped with an acceleration sensor, the algorithm can be further extended by implementing direct measurement of reproduced acceleration. A simulation has been performed in this context, where white noise is added to the output measured reproduced acceleration value (see Fig. 4). As the Figure shows, the algorithm is capable of correctly reproducing the required acceleration value while respecting the applicable constraints even in this scenario. Once direct measurement of reproduced acceleration is available, the motion platform's dynamic properties can be included in the control loop; an ideal system has been assumed in the design phase.

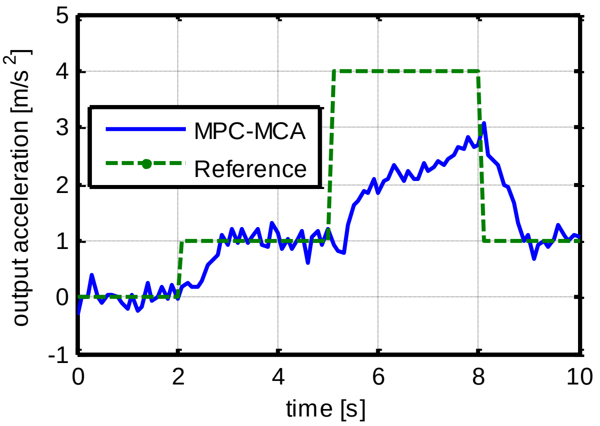

Figure 4: Simulation with white-noise added to the output signal

CONCLUSION

This paper describes the use of an MPC algorithm to control and optimize MCA. Classic MCAs have to be set up so as to ensure that the relevant kinematic limits are not exceeded even in a worst-case scenario, resulting in the platform's potential not being used to the full extent. The paper describes a way of enhancing MCA with MPC feedback control – a solution capable of using the full kinematic potential of the motion platform while providing the optimum results with respect to the given constraints. All theoretical results have been verified during simulations, including the proposed algorithm capable of dynamically adjusting the constraints. In this context, follow-up work will be focused mainly on implementing the algorithm in an actual simulator.

AUTHOR BIOGRAPHY

EVŽEN THÖNDEL was born in Prague, the Czech Republic, and studied at the Czech Technical University (CTU) in Prague. In 2004, he acquired a Master's Degree in Technical Cybernetics, and in 2008 a Ph.D. in electrical engineering and materials. Today, he works at the CTU as lecturer. Since 2009, Evžen has been working for Pragolet as simulation technology development specialist.

REFERENCES

Domagoj, J. and Budin, L. 2002. "Forward Kinematics of a Stewart Platform Mechanism". Faculty of Electrical Engineering and Computing.

Fang, Z. and Kemeny, A. 2012. "Explicit MPC Motion Cueing Algorithm for Real-Time Driving Simulator". In IEEE 7th International Power Electronics and Motion Conference, Harbin, China, 874-878.

Groen, J. J. and Jongkees, L. B. W. 1948. "The Threshold of Angular Acceleration Perception". The Journal of Physiology, 107, 1-7.

Havlena, V. and Štecha, J. 2002. "Teorie dynamických systémů". Czech Technical University in Prague, ISBN 80-01-01971-3.

Jamson, A. H. J. 2010. "Motion Cueing in Driving Simulators for Research Applications". Ph.D. thesis, The University of Leeds, Institute for Transport Studies.

MathWorks, 2013. "Model Predective Control Toolbox". http://www.mathworks.com/help/mpc/.

Telban, R. J.; Wu, W. and Cardullo, F. M. 2000. "Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithm". Technical report, State University of New York.

Thöndel, E. 2012. "Design and Optimisation of a Motion Cueing Algorithm for a Truck Simulator". In proceeding of the 2012 European simulation and modelling conference, Essen, Germany, 165-170.

Thöndel, E. 2011. "Design and Optimal Control of a Linear Electromechanical Actuator for Motion Platforms with Six Degrees of Freedom". Intelligent Automation and Systems Engineering, Springer, 65-77.