Future Reference Prediction in Model Predictive Control based Driving Simulators

Arash Mohammadi, Houshyar Asadi, Shady Mohamed, Kyle Nelson, Saeid Nahavandi
Institute for Intelligent Systems Research and Innovation (IISRI)
Deakin University, Victoria 3216, Australia
arash.m@research.deakin.edu.au, houshyar.asadi@deakin.edu.au,
shady.mohamed@deakin.edu.au, kyle.nelson@deakin.edu.au, saeid.nahavandi@deakin.edu.au

Abstract

The goal of a driving simulator is to produce an environment for a driver similar to the real driving scenario. Motion cueing algorithms are used to produce a realistic motion while respecting the workspace limitations and motion simulator boundaries. Model Predictive Control has become popular recently for motion cueing. However, in this control method, the optimization is based on a predefined constant future input trajectory while it is not a practical assumption. In this research, a method is proposed to predict the future reference based on the finite history of input. This method does not require the position trajectory to follow a specific road. Simulation results show the effectiveness of the proposed model predictive control method in terms of realistic motion sensation for a driver.

1 Introduction

Driving simulators are safe and cost effective tools to mimic a real world driving environment via producing similar motion, vision and audio [Chang *et al.*, 2009]. Driving simulators have wide applications such as training, prototyping, testing advanced driving assistance and driver behavioural study [Lee *et al.*, 2015; 1998]. The inclusion of motion cues improves feeling of a real driving and helps the driver to have a natural reaction [Houck *et al.*, 2005; Garrett and Best, 2013; Siegler *et al.*, 2001].

One of the main limitations of driving simulators is their limited workspace. Therefore, it is impossible for a motion simulator to generate a realistic motion. A Motion Cueing Algorithm (MCA) is used to produce a high fidelity motion while maintaining the motion platform within the simulator limitations. Hence a desired motion can be converted into a feasible motion for a driving simulator. Despite technological advances in different aspects of motion simulators, MCA remains as one of the main challenges [Houck *et al.*, 2005; Lee *et al.*, 2015; Fang *et al.*, 2014].

Producing a correct motion by driving simulator is important as mismatches between motion and visual cues (false motion cues) lead to motion sickness [Slob, 2008]. Feeling disorientation, vertigo, dryness of mouth, cold sweating and blurred vision are symptoms of motion sickness [Gable and Walker, 2013; Maran, 2013; Warren and Wertheim, 2014]. Thus, false motion cues have to be avoided by providing an effective MCA.

The main human motion sensor known as vestibular system is located in the inner ear and made of otoliths and semicircular canals responsible for detection of linear and rotational accelerations respectively [Houck *et al.*, 2005; Fang *et al.*, 2014; Pretto *et al.*, 2015; Grant and Reid, 1997; Asadi *et al.*, 2016]. Since otoliths cannot discriminate between gravity and sustained acceleration, a technique called tilt-coordination is used in MCA to produce sustained acceleration by tilting a driver with a slow rate below human rotational perception threshold to avoid rotational motion detection but feeling sustained acceleration [Fang *et al.*, 2014; Lee *et al.*, 2015; Chapron and Colinot, 2007].

Originally, washout filters have been used for MCA. The classical washout filters are made of high-pass and low-pass filters. These filters separate onset and sustained accelerations. The onset acceleration feeling is produced via accelerating the driver linearly while sustained accelerations feeling is produced via tilt-coordination [Conrad *et al.*, 1973; Asadi *et al.*, 2014b; 2014a]. Classical washout filters have been improved by applying Linear Quadratic Regulator (LQR) method and called optimal washout filters [Houck *et al.*, 2005; Wu and Cardullo, 1997; Sivan *et al.*, 1982].

The main drawback of washout filters is that they are unable to consider the physical boundaries of motion simulator hence they are tuned for the worst case scenario and they cannot provide the best performance. Therefore, in recent years, a new type of MCA based on Model Predictive Control (MPC) has received a huge attention.

MPC is a multi-variable control method based on constrained optimization with popular applications in academia and industries [Bemporad, 2016; Ferreau, 2006; Hovd, 2004; Romero *et al.*, 2015]. In MPC control methods, a sequence of control actions are derived over a finite future horizon considering the explicitly formulated process model to predict and achieve the optimized future behaviour. [Kumar and Ahmad, 2012; Kouramas *et al.*, 2011]

MPC in MCA has been proposed first by Dagdelen *et al.* [Dagdelen *et al.*, 2004; 2009]. Augusto [Augusto and Loureiro, 2009] applied tilt-coordination and a vestibular model in MPC-based MCA. As the input from driver to motion simulator is not known beforehand, at each moment he assumed the input reference will remain constant during prediction horizon for MPC optimization. He also developed an MPC-MCA with assumption that the rate of input will remain constant, however, after comparing the results he concluded that considering the reference as constant is a better option.

2 MPC-based MCA

Model Predictive Control is a control method which optimizes the output of a system over a predicted future period of time called prediction horizon (N_pT_s) via applying an optimized control sequence over a control horizon (N_cT_s). The model is discretised with sample time T_s. This method explicitly uses the model of a system to minimize a cost function. The prediction and control horizons recede at each sampling time and the control sequence is re-optimized. This control method is able to handle multi-input multi-output systems and the constraints on variables are respected during the process of optimization [Camacho and Alba, 2013; Morari *et al.*, 2014; Hrovat *et al.*, 2012]. Figure 1 illustrates the schematic diagram for model predictive control horizons.

In MPC-MCA, the strictly proper discrete state space representation of the system is extracted

$$\begin{aligned}\mathbf{x}_m(t+1) &= \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}_m\mathbf{x}_m(t)\end{aligned} \tag{1}$$

A new variable $\Delta\mathbf{u}$ is defined to apply constraints on input rates, input and output values

$$\Delta\mathbf{u}(t) \triangleq \mathbf{u}(t) - \mathbf{u}(t-1) \tag{2}$$

$$\Delta\mathbf{x}_m(t) \triangleq \mathbf{x}_m(t) - \mathbf{x}_m(t-1) \tag{3}$$

The augmented state is defined as

$$\mathbf{x}(t) \triangleq \begin{bmatrix}\Delta\mathbf{x}_m(t)\\ \mathbf{y}(t)\end{bmatrix} \tag{4}$$

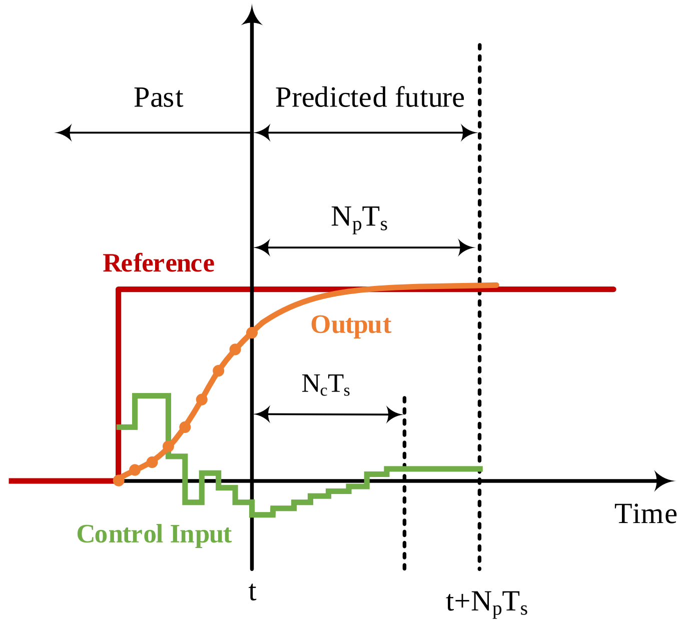

Figure 1: Schematic diagram of model predictive control horizons

According to new variables

$$\begin{aligned}\mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\Delta\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)\end{aligned} \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix}\mathbf{A}_m & \mathbf{0}\\ \mathbf{C}_m\mathbf{A}_m & \mathbf{I}\end{bmatrix}, \mathbf{B} = \begin{bmatrix}\mathbf{B}_m\\ \mathbf{C}_m\mathbf{B}_m\end{bmatrix}, \mathbf{C} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix} \tag{6}$$

In Equation 5, $\Delta\mathbf{u}(t)$ is the control variable and $\mathbf{x}(t)$ is the state variable.

At time t, the predicted state variable of n^{th} next sample time $(t+nT_s)$ is shown by $\mathbf{x}(t+n|t)$. Similarly, the predicted output of time $(t+nT_s)$ at t is shown by $\mathbf{y}(t+n|t)$ as follows

$$\begin{aligned}&\mathbf{x}(t+n|t) = \mathbf{A}^n\mathbf{x}(t) + \\ &\Sigma_{i=1}^{\min(N_c,n)}\mathbf{A}^{N_p-i}\mathbf{B}\Delta\mathbf{u}(t+i-1)\end{aligned} \tag{7}$$

$$\begin{aligned}&\mathbf{y}(t+n|t) = \mathbf{C}\mathbf{A}^n\mathbf{x}(t+n|t) + \\ &\Sigma_{i=1}^{\min(N_c,n)}\mathbf{C}\mathbf{A}^{N_p-i}\mathbf{B}\Delta\mathbf{u}(t+i-1)\end{aligned} \tag{8}$$

After defining $\mathbf{Y}$ and $\Delta\mathbf{U}$,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}(t+1|t) \\ \mathbf{y}(t+2|t) \\ \vdots \\ \mathbf{y}(t+N_p|t) \end{bmatrix} \in \mathbb{R}^{(N_p N_{out})\times 1} \tag{9}$$

$$\Delta\mathbf{U} = \begin{bmatrix} \Delta\mathbf{u}(t) \\ \Delta\mathbf{u}(t+1) \\ \vdots \\ \Delta\mathbf{u}(t+N_c-1) \end{bmatrix} \in \mathbb{R}^{(N_c N_{in})\times 1} \tag{10}$$

the relationship between $\mathbf{Y}$, $\mathbf{X}$, $\Delta\mathbf{U}$ follows

$$\mathbf{Y} = \mathbf{F}\mathbf{X}(t) + \mathbf{\Phi}\Delta\mathbf{U} \tag{11}$$

where $\mathbf{F}$ and $\mathbf{\Phi}$ are

$$\mathbf{F} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \vdots \\ \mathbf{CA}^{N_p} \end{bmatrix} \tag{12}$$

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0} \\ \mathbf{CA}^2\mathbf{B} & \mathbf{CAB} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N_p-1}\mathbf{B} & \mathbf{CA}^{N_p-2}\mathbf{B} & \dots & \mathbf{CA}^{N_p-N_c}\mathbf{B} \end{bmatrix}$$

The main target is to minimize the following cost function

$$J_1(\Delta\mathbf{U}) = (\mathbf{Y}_{ref} - \mathbf{Y})^\intercal \mathbf{Q}(\mathbf{Y}_{ref} - \mathbf{Y}) + \mathbf{U}^\intercal\mathbf{S}\mathbf{U} + \Delta\mathbf{U}^\intercal\mathbf{R}\Delta\mathbf{U} \tag{13}$$

where $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{S}$ are the weighting matrices for error, input rate and input respectively.

For further calculations, matrices $\mathbf{U}$, $\mathbf{T}$ and $\bar{\mathbf{U}}_i$ are defined as follows

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{u}(t+1) \\ \vdots \\ \mathbf{u}(t+N_c-1) \end{bmatrix} \tag{14}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix} \tag{15}$$

$$\bar{\mathbf{U}}_i = \begin{bmatrix} \mathbf{u}(t-1) \\ \mathbf{u}(t-1) \\ \vdots \\ \mathbf{u}(t-1) \end{bmatrix} \tag{16}$$

The relationship between input and input variation is as

$$\mathbf{U} = \mathbf{T}\Delta\mathbf{U} + \bar{\mathbf{U}}_i \tag{17}$$

After converting input sequence to the variation of input sequence and removing the constant terms from cost function, the objective is to minimize

$$J(\Delta\mathbf{U}) = \Delta\mathbf{U}^\intercal(\mathbf{\Phi}^\intercal\mathbf{Q}\mathbf{\Phi} + \mathbf{R} + \mathbf{T}^\intercal\mathbf{S}\mathbf{T})\Delta\mathbf{U} + 2\Delta\mathbf{U}^\intercal(\mathbf{\Phi}^\intercal\mathbf{Q}(\mathbf{F}\mathbf{x}(t) - \mathbf{Y}_{ref}) + \mathbf{T}^\intercal\mathbf{S}\bar{\mathbf{U}}_i) \tag{18}$$

which can be rewritten in a short form as follows

$$J(\Delta\mathbf{U}) = \frac{1}{2}\Delta\mathbf{U}^\intercal\mathbf{H}\Delta\mathbf{U} + \Delta\mathbf{U}^\intercal\mathbf{g} \tag{19}$$

The constraints can be represented as

$$\mathbf{n}_{\min} \leq \mathbf{M}\Delta\mathbf{U} \leq \mathbf{n}_{\max} \tag{20}$$

The constraints can be split into three categories representing constraints on input rates, input values and output results

$$\begin{bmatrix} \mathbf{n}_{1\,\min} \\ \mathbf{n}_{2\,\min} \\ \mathbf{n}_{3\,\min} \end{bmatrix} \leq \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \mathbf{M}_3 \end{bmatrix} \Delta\mathbf{U} \leq \begin{bmatrix} \mathbf{n}_{1\,\max} \\ \mathbf{n}_{2\,\max} \\ \mathbf{n}_{3\,\max} \end{bmatrix} \tag{21}$$

Both Equations 19 and 20 form a Quadratic Programming (QP) problem. Several methods are available for solving QP problems [Gill and Wong, 2015]. This optimization has to be recalculated at each sampling time. Once the optimized control input is obtained, only the control input of the first sample is applied and the rest of control inputs are discarded for the re-optimisations at the next sample times.

As there is no clear estimation of Y_{ref}, each output reference is usually considered constant during the entire prediction horizon. This unrealistic assumption prevents MPC from providing its best result.

3 Proposed method

In this research, we have applied an MPC-MCA with a prediction of future output reference based on its finite history. An Artificial Neural Network (ANN) is considered for being trained according to a set of driving records.

The training reference is sampled with a uniform sampling time, hence it can be represented by the following sequence

$$u_1, u_2, \dots, u_N \tag{22}$$

where u_i represents reference at sample i and N represents the total number of samples. For each sample i, the corresponding finite history is

$$u_{i-n_h}, u_{i-n_h+1}, \dots, u_{i-1}, u_i \tag{23}$$

and the corresponding finite future is

$$u_{i+1}, u_{i+2}, \dots, u_{i+n_f-1}, u_{i+n_f} \tag{24}$$

where n_h is the number of samples for history and n_f is the number of future samples.

The past samples are divided into several regions with sample numbers n_{h1} to n_{h5} where

$$n_{h1} + n_{h2} + n_{h3} + n_{h4} + n_{h5} = n_h \tag{25}$$

For comfortable indexing, we define cumulative sum of regions numbers

$$N_{h,m} = \Sigma_{k=1}^{m} n_{hk}, \quad \forall m \in \{0, 1, 2, 3, 4, 5\} \tag{26}$$

Each region set is defined as

$$U_{h,m}(i) = \{u_k | i + N_{h,m-1} < k < i + N_{h,m}\} \tag{27}$$

Figure 2 illustrates the region for past and future of an input signal at sample time i.

The average of each part is considered as a representative of their set

$$\bar{u}_{h1}, \bar{u}_{h2}, \bar{u}_{h3}, \bar{u}_{h4}, \bar{u}_{h5} \tag{28}$$

where

$$\bar{u}_{h,m}(i) = \frac{1}{n_{hm}} \Sigma U_{h,m}(i), \quad \forall m \in \{1, 2, 3, 4, 5\} \tag{29}$$

Similarly, several regions with n_{fm} samles are considered for future

$$n_{f1} + n_{f2} + n_{f3} + n_{f4} + n_{f5} = n_f \tag{30}$$

$$N_{f,m} = \Sigma_{k=1}^{m} n_{f,k}, \quad \forall m \in \{0, 1, 2, 3, 4, 5\} \tag{31}$$

The future regions are

$$U_{f,m}(i) = \{u_k | i + N_{f,m-1} < k < i + N_{f,m}\} \tag{32}$$

Each future set is represented by its average

$$\bar{u}_{f,m}(i) = \frac{1}{n_{fm}} \Sigma U_{f,m}(i), \quad \forall m \in \{1, 2, 3, 4, 5\} \tag{33}$$

The quintuples made of $\bar{u}_{h,m}$ for each sampling time are the inputs to neural network and the quintuples made of future averages $\bar{u}_{f,m}$ are the expected output for training.

The past and future regions are considered with variable width where higher resolution is for regions closer to the current sampling time.

$$n_{h,m-1} = 1.5 \times n_{h,m}, \quad \forall m \in \{2, 3, 4, 5\} \tag{34}$$
$$n_{f,m} = 1.5 \times n_{f,m+1}, \quad \forall m \in \{1, 2, 3, 4\} \tag{35}$$

On the test signal, the history of input is given to the neural network and the output of neural network is used for generation of an approximation about the future reference signal. At each sampling time, this future reference is updated and used by MPC for optimization of control input to motion simulator.

4 Results and discussion

In this section, the results of the proposed method for output reference prediction are presented. A training set made of 49262 inputs is prepared for the neural network. MATLAB software is used for training this data. A hidden layer with size of 36 is applied according to figure 3. The network is trained and prepared for responding to test inputs. Figure 4 shows a single frame of a future reference predicted by the neural network. As future input signal is uncertain, the predicted future signal can be a realistic or very far from the real future. However, in general it performs better than assuming the predefined constant future signal for the entire prediction horizon. The predicted future signal has to be updated at each sample time via neural network. As the output is identified at 5 spots, their interpolations are used in the middle points.

An MPC with control horizon 3, predictive horizon 400 and sampling time $10ms$ has been applied for a

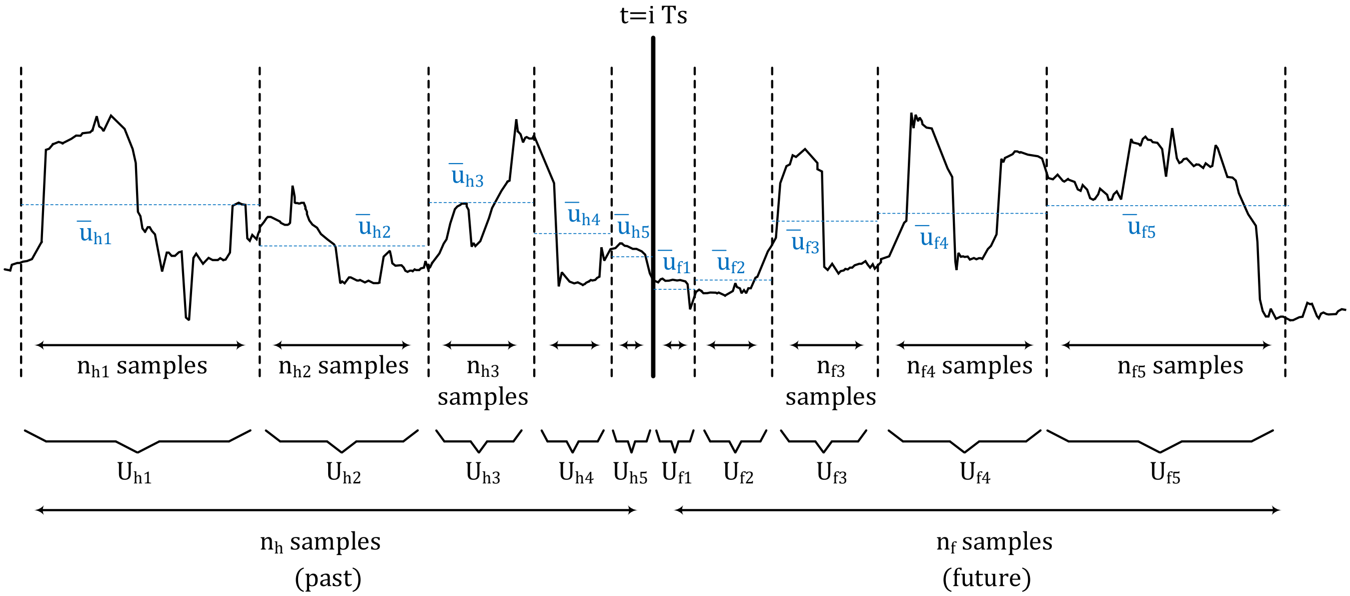

Figure 2: Schematic of region division for a reference signal

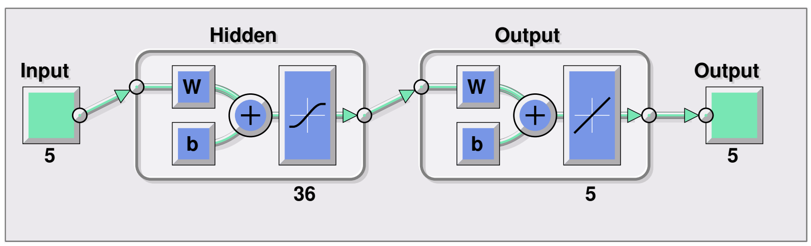

Figure 3: The applied neural network for future reference prediction in the proposed MPC-MCA

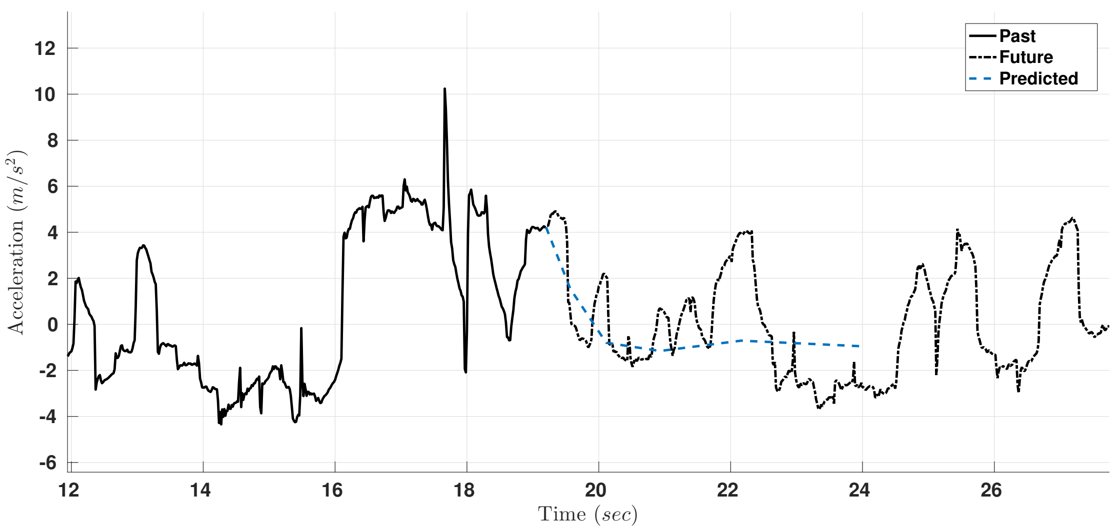

Figure 4: Prediction of future acceleration reference signal

surge acceleration input. Figures 5 demonstrates the sensed specific force by the driver according to the specific force and human vestibular system model of Telban [Houck *et al.*, 2005]. The reference sensed specific force is the force that a driver feels in a real world vehicle. A desired sensed motion has to follow the reference output signal accurately. A comparison between the sensed motion signals shows that the overall sensed specific force is improved in the proposed method compared to when the output reference is predefined. The difference between the sensed motion from the real world driver and motion simulator driver is called sensation error. The Root Mean Square (RMS) of sensation error is decreased in the proposed method from $1.29\frac{m}{s^2}$ to $0.85\frac{m}{s^2}$.

Figure 6 shows the displacement of motion platform for generating the required motion for the simulator driver. In the proposed method the maximum displacement has increased from $1.35m$ to $2.78m$. The proposed method is respecting the adjusted linear displacement limitation of $\pm 3m$. It should be noted that the proposed method behaves less conservative in terms of workspace usage. This higher utilization of displacement was conducted to reduce the output sensation error. As in MPC with constant future reference a high peak of acceleration is predicted to be constant for the entire prediction horizon, the system is considered to be challenging while a high peak of acceleration does not practically last for a long time. Therefore, the previous MPC performed conservatively and inefficient. The simulation results show supremacy of the proposed method over previous MPC with constant output reference due to using ANN and better knowledge about the reference.

5 Conclusion

In MPC-based MCA, the MPC reference is typically assumed to be constant. In this research, a neural network method is applied to predict the future of input acceleration signal according to its finite history. This method does not require the driver to follow a specific route. The simulation results demonstrate that MPC-MCA with proposed future reference estimation has better followed the sensation of a driver in a real vehicle. The proposed MPC is less conservative than traditional MPC with predefined constant reference and uses the workspace efficiently.

References

[Asadi *et al.*, 2014a] Houshyar Asadi, Arash Mohammadi, Shady Mohamed, and Saeid Nahavandi. Adaptive translational cueing motion algorithm using fuzzy based tilt coordination. In *International Conference on Neural Information Processing*, pages 474–482. Springer, 2014.

[Asadi *et al.*, 2014b] Houshyar Asadi, Arash Mohammadi, Shady Mohamed, Delpak Rahim Zadeh, and Saeid Nahavandi. Adaptive washout algorithm based fuzzy tuning for improving human perception. In *International Conference on Neural Information Processing*, pages 483–492. Springer, 2014.

[Asadi *et al.*, 2016] Houshyar Asadi, Shady Mohamed, Chee Peng Lim, and Saeid Nahavandi. A review on otolith models in human perception. *Behavioural brain research*, 309:67–76, 2016.

[Augusto and Loureiro, 2009] Bruno Augusto and Rui Loureiro. Motion cueing in the chalmers driving simulator-a model predictive control approach. 2009.

[Bemporad, 2016] Alberto Bemporad. A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control. *IEEE Transactions on Automatic Control*, 61(4):1111–1116, 2016.

[Camacho and Alba, 2013] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[Chang *et al.*, 2009] Yang-Hung Chang, Chung-Shu Liao, and Wei-Hua Chieng. Optimal motion cueing for 5-dof motion simulations via a 3-dof motion simulator. *Control Engineering Practice*, 17(1):170–184, 2009.

[Chapron and Colinot, 2007] Thomas Chapron and Jean-Pierre Colinot. The new psa peugeot-citroen advanced driving simulator overall design and motion cue algorithm. In *Proceedings of Driving Simulation Conference*, 2007.

[Conrad *et al.*, 1973] BJORN Conrad, SF Schmidt, and JG Douvillier. Washout circuit design for multi-degrees of freedom moving base simulators. In *Proceedings of the AiAA Visual and Motion Simulation Conference, Palo Alto (CA), September 10*, volume 12, 1973.

[Dagdelen *et al.*, 2004] M Dagdelen, G Reymond, A Kemeny, M Bordier, and N Maïki. Mpc based motion cueing algorithm: Development and application to the ultimate driving simulator. In *Conférence simulation de conduite*, pages 221–233, 2004.

[Dagdelen *et al.*, 2009] Mehmet Dagdelen, Gilles Reymond, Andras Kemeny, Marc Bordier, and Nadia Maïzi. Model-based predictive motion cueing strategy for vehicle driving simulators. *Control Engineering Practice*, 17(9):995–1003, 2009.

[Fang *et al.*, 2014] Zhou Fang, Florent Colombet, Jean-Christophe Collinet, and Andras Kemeny. Roll tilt thresholds for 8 dof driving simulators. 2014.

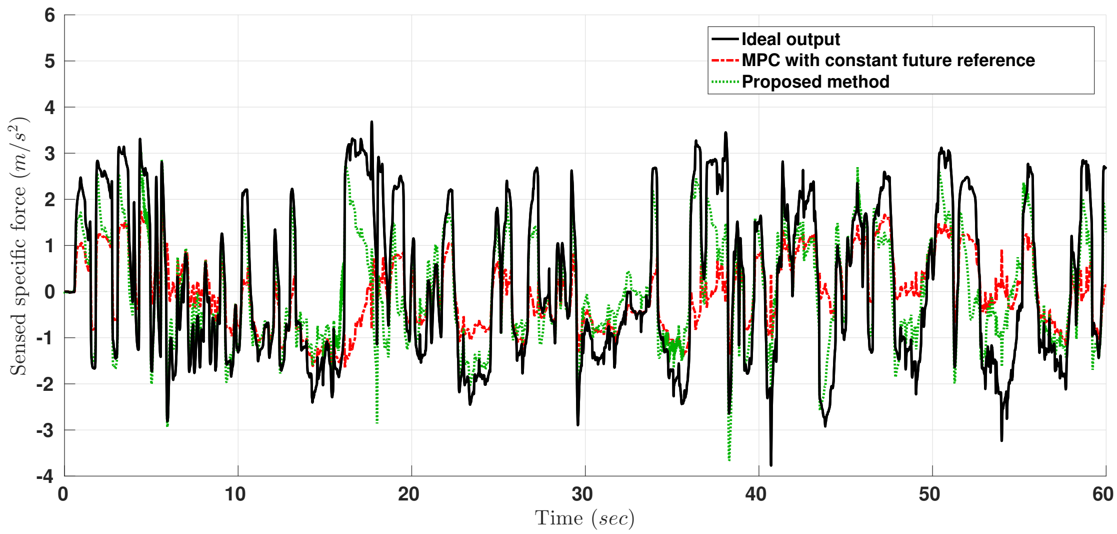

Figure 5: Comparison of sensed specific force

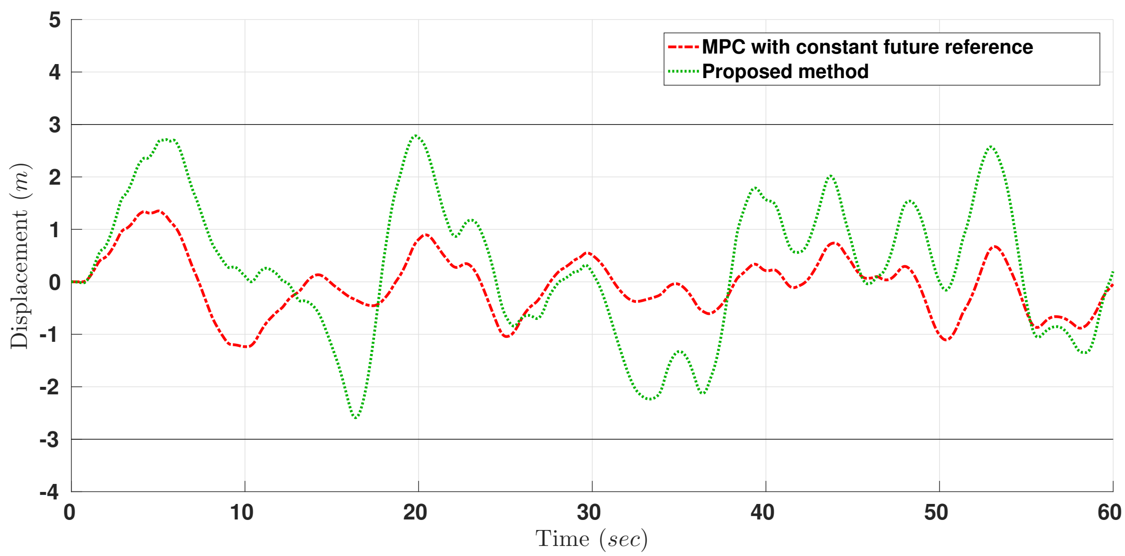

Figure 6: Comparison of Displacement

[Ferreau, 2006] Hans Joachim Ferreau. An online active set strategy for fast solution of parametric quadratic programs with applications to predictive engine control. *University of Heidelberg*, 2006.

[Gable and Walker, 2013] Thomas M Gable and Bruce N Walker. Driving simulator sickness screening: Efficient and effective new protocol. In *Proceedings of the Fifth International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2013)*, pages 101–102, 2013.

[Garrett and Best, 2013] Nikhil JI Garrett and Matthew C Best. Model predictive driving simulator motion cueing algorithm with actuator-based constraints. *Vehicle System Dynamics*, 51(8):1151–1172, 2013.

[Gill and Wong, 2015] Philip E Gill and Elizabeth Wong. Methods for convex and general quadratic programming. *Mathematical Programming Computation*, 7(1):71–112, 2015.

[Grant and Reid, 1997] Peter R Grant and Lloyd D Reid. Motion washout filter tuning: Rules and requirements. *Journal of Aircraft*, 34(2):145–151, 1997.

[Houck *et al.*, 2005] Jacob A Houck, Robert J Telban, and Frank M Cardullo. Motion cueing algorithm development: Human-centered linear and nonlinear approaches. 2005.

[Hovd, 2004] Morten Hovd. A brief introduction to model predictive control. *URL= http://www. itk. ntnu. no/fag/TTK4135/viktig/MPCkompendium%20HOvd. pdf*, 2004.

[Hrovat *et al.*, 2012] Davor Hrovat, Stefano Di Cairano, H Eric Tseng, and Ilya V Kolmanovsky. The development of model predictive control in automotive industry: A survey. In *Control Applications (CCA), 2012 IEEE International Conference on*, pages 295–302. IEEE, 2012.

[Kouramas *et al.*, 2011] Konstantinos I Kouramas, Nuno P Faísca, C Panos, and Efstratios N Pistikopoulos. Explicit/multi-parametric model predictive control (mpc) of linear discrete-time systems by dynamic and multi-parametric programming. *Automatica*, 47(8):1638–1645, 2011.

[Kumar and Ahmad, 2012] A Senthil Kumar and Zainal Ahmad. Model predictive control (mpc) and its current issues in chemical engineering. *Chemical Engineering Communications*, 199(4):472–511, 2012.

[Lee *et al.*, 1998] Woon-Sung Lee, Jung-Ha Kim, and Jun-Hee Cho. A driving simulator as a virtual reality tool. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 71–76. IEEE, 1998.

[Lee *et al.*, 2015] Seong-Whan Lee, Heinrich H Bülthoff, and Klaus-Robert Müller. *Recent Progress in Brain and Cognitive Engineering*, volume 5. Springer, 2015.

[Maran, 2013] Fabio Maran. *Model-Based Control Techniques*. PhD thesis, University of Padova, 2013.

[Morari *et al.*, 2014] Manfred Morari, Carlos E Garcia, and David M Prett. Model predictive control: theory and practice. In *Proc./FAC Workshop on Model Based Process Control*, pages 1–12, 2014.

[Pretto *et al.*, 2015] Paolo Pretto, Joost Venrooij, Alessandro Nesti, and Heinrich H Bülthoff. Perception-based motion cueing: A cybernetics approach to motion simulation. In *Recent Progress in Brain and Cognitive Engineering*, pages 131–152. Springer, 2015.

[Romero *et al.*, 2015] Alberto Romero, Dean Millar, Monica Carvalho, José M Maestre, and Eduardo F Camacho. A comparison of the economic benefits of centralized and distributed model predictive control strategies for optimal and sub-optimal mine dewatering system designs. *Applied Thermal Engineering*, 90:1172–1183, 2015.

[Siegler *et al.*, 2001] I Siegler, G Reymond, A Kemeny, and A Berthoz. Sensorimotor integration in a driving simulator: contributions of motion cueing in elementary driving tasks. In *Proceedings of Driving Simulation Conference*, pages 21–32, 2001.

[Sivan *et al.*, 1982] Raphael Sivan, Jehuda Ish-Shalom, and Jen-Kuang Huang. An optimal control approach to the design of moving flight simulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 12(6):818–827, 1982.

[Slob, 2008] JJ Slob. State-of-the-art driving simulators, a literature survey. *DCT Report*, 107, 2008.

[Warren and Wertheim, 2014] Rik Warren and Alexander H Wertheim. *Perception and Control of Self-motion*. Psychology Press, 2014.

[Wu and Cardullo, 1997] Weimin Wu and Frank M Cardullo. Is there an optimum motion cueing algorithm. In *Proceedings of the AIAA Modelling and Simulation Technologies Conference, New Orleans (LA), August 11*, volume 13, pages 23–29, 1997.