# A Frequency-Aware Model Predictive Control Motion Cueing Algorithm

**Maurice Kolff[1], Robert Jacumet[2,3], Sebastian Wagner[2], Dirk Wollherr[3], Marion Leibold[3]**

***Abstract -*** *This paper presents a Model Predictive Control framework for driving simulator motion cueing including effective control of the frequency-domain characteristics of the simulator motion. By transforming the predicted output sequence using the Discrete Fourier Transform (DFT) matrix method, the controller can penalize or amplify specific frequency components. The method is first demonstrated using a discrete-frequency multisine reference use-case, followed by a realistic multi degree-of-freedom driving simulation use-case. Although its inclusion adds additional constraints on the prediction horizon and the simulation sample time, the frequency-aware motion cueing algorithm enhances the frequency response in both use-cases, thus providing an effective control over the frequency characteristics of the simulator motion. This leads to a better integration of frequency-dependent motion characteristics, a more effective use of the simulator motion workspace, and is expected to provide an improved human perception of the motion.*

(1) Delft University of Technology, Faculty of Mechanical Engineering, section Intelligent Vehicles, 2628 CD Delft, e-mail : m.j.c.kolff@tudelft.nl
(2) BMW Group, Research, New Technologies, Innovations, 80788 Munich, e-mail : {robert.jacumet, sebastian.ws.wagner}@bmw.de
(3) Technical University of Munich, School of Computation, Information and Technology, Chair of Automatic Control Engineering, 82090 Munich, e-mail : {r.jacumet, dirk.wollherr, marion.leibold}@tum.de

***Keywords:*** *Motion Cueing Algorithm, Model Predictive Control, Frequency Tracking, Discrete Fourier Transform*


https://doi.org/10.82157/dsa/2025/16

## 1. Introduction

Over the last decade, Model-Predictive Control (MPC) has gained significance in Motion Cueing Algorithms (MCAs) for driving simulators (Dagdelen, et al., 2009). With the ability to predict the simulator system behavior over a defined horizon, MPC allows for the generation of simulator motion cues that maximize perceptual fidelity while adhering to simulator platform limitations (Bruschetta, Maran, and Beghi, 2017b). The recent surge in MPC applications has been driven by advancements in computational power and the increasing complexity of motion simulators, which demand more sophisticated algorithms to deliver realistic driving experiences compared the traditional filter-based approaches. In fact, MPC-based MCAs have been shown to allow for a better motion cueing quality, i.e., a higher perceived level of realism, compared to filter-based approaches (Biemelt, et al., 2021; Cleij, et al., 2019).

A critical aspect of motion cueing quality, however, is the frequency of the generated motion cues, because the vestibular system is highly sensitive to specific frequency ranges (Telban, Cardullo, and Kelly, 2005). Therefore, ensuring the correct balance, as well as the consistency of frequency components can be essential for the motion simulation. Although the relevance of frequency components is acknowledged in filter-based MCAs (Ellensohn, et al., 2019), MPC-based MCAs optimize the motion purely in the time domain, eventually leading to degraded performance in certain frequency bands. It would thus be highly beneficial to extend the existing MPC formulations, such that the frequency response of the simulator motion can be directly controlled. Whereas frequency-aware MPC has been proposed before (Grandia, et al., 2019; Stickan, et al., 2019), and the Fourier transform has been introduced in driving simulation to improve the MPC performance (Fang, Wautier, and Kemeny, 2022), to the best of our knowledge, an actual MPC formulation including frequency control within the context of driving simulation does not yet exist.

To address this challenge, the present paper presents a method to incorporate the frequency response of the motion cues directly into the cost function of an MPC. This is achieved by applying the Discrete Fourier Transform (DFT) in matrix form in the MPC cost function. By transforming the time-domain motion cueing signals into the frequency domain, the DFT matrix enables the explicit representation and control of frequency components in the MPC opti-

mization process. This includes the possibility to provide consistency between the reference and simulator motion in terms of their frequency response. Furthermore, certain frequency bands can be specifically avoided or prioritized. Apart from its mathematical structure, the present paper also provides simulation results of two use-cases.

The paper is structured as follows. Section 2 provides the mathematical formulation for the incorporation of frequency control in an MPC algorithm. Section 3 provides the simulation results. Section 4 discusses implications, limitations, and future work. Finally, the paper is concluded in Section 5.

# 2. Methods

This section presents how the DFT is incorporated in an MPC-based MCA formulation to efficiently control the resulting frequencies. First, the cost function is appended with a frequency-domain tracking term. Then, the resulting MPC is reformulated as a quadratic program (QP) for which a range of fast and reliable real-time solvers exists (Freund, 2004; Wang and Boyd, 2010).

## 2.1. Frequency-Aware Cost Function

A simplified typical linear MPC MCA formulation is given as (Beghi, Bruschetta, and Maran, 2012; Bruschetta, Maran, and Beghi, 2017a; Qazani, Asadi, and Nahavandi, 2020)

$$J_t = \min_{\boldsymbol{U}} \sum_{n=0}^{H_p-1} (\boldsymbol{y}_{S,n} - \boldsymbol{y}_{S,n}) \boldsymbol{Q}_t (\boldsymbol{y}_{S,n} - \boldsymbol{y}_{S,n}) + \boldsymbol{u}_n \boldsymbol{R} \boldsymbol{u}_n \tag{1a}$$

$$\text{s.t. for } n = 0, \ldots, H_p - 1:$$

$$\boldsymbol{x}_{n+1} = \boldsymbol{A}\boldsymbol{x}_n + \boldsymbol{B}\boldsymbol{u}_n, \tag{1b}$$

$$\boldsymbol{y}_{S,n} = \boldsymbol{C}\boldsymbol{x}_n + \boldsymbol{D}\boldsymbol{u}_n, \tag{1c}$$

$$\boldsymbol{x} \in \mathcal{X}, \quad \boldsymbol{u} \in \mathcal{U}, \tag{1d}$$

with $\boldsymbol{y}_V$ the vehicle motion, $\boldsymbol{y}_S$ the simulator motion, $\boldsymbol{U} = (\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{H_p-1})$, $n$ the prediction step, $H_p$ the prediction horizon, and $\boldsymbol{Q}_t$ and $\boldsymbol{R}$ the weighting matrices of the respective terms. Moreover, the system state $\boldsymbol{x}$ and the simulator control inputs $u$ are bound by the sets $\mathcal{X}$ and $\mathcal{U}$, respectively. The cost function $J_t$ in (1) solely considers the time domain behavior.

To consider the frequencies of a real, time domain signal $y$ in the cost function, we take the DFT:

$$Y_k = \sum_{j=0}^{N-1} y_j e^{-i\frac{2\pi}{N}kj}, \quad k = 0, 1, \ldots, N-1.$$

with the complex number $i$ and $Y_k \in \mathbb{C}^N$. A typical numerical approach is to use the Fast Fourier Transform (FFT). As numerical quadratic solvers generally require a description of the system in matrix form, directly employing the FFT in the optimization scheme is not feasible. Instead, it is possible to calculate the DFT using matrix multiplication (Stickan, et al., 2019), such that:

$$Y_k = W_{k,j} y_j, \tag{2}$$

with $Y$ the DFT of $y$ and where $\boldsymbol{W}$ is the DFT matrix, defined as:

$$\boldsymbol{W} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \Omega & \Omega^2 & \cdots & \Omega^{N-1} \\ 1 & \Omega^2 & \Omega^4 & \cdots & \Omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \Omega^{N-1} & \Omega^{2(N-1)} & \cdots & \Omega^{(N-1)^2} \end{bmatrix}, \tag{3}$$

with $N$ being the total number of samples, $\boldsymbol{W} = \mathbb{C}^{N \times N}$ and $\Omega$ being the primitive $N$th root of unity:

$$\Omega = e^{2\pi i/N}. \tag{4}$$

When applying the DFT in the context of MPC, the number of samples is most conveniently set equal to the number of prediction steps, i.e., $N = H_p$. As long as the prediction horizon $H_p$ is constant, the DFT matrix $\boldsymbol{W}$ will also be constant and thus does not have to be calculated at each iteration step.

When performing the DFT of a real signal, the frequency components $Y_k$ are obtained for the positive frequency bins

$$f_k = \frac{k}{N} f_s, \qquad k = 0,\ 1,\ \ldots,\ \left\lfloor \frac{N}{2} \right\rfloor, \tag{5}$$

and, hence, depend on the sample size $N = H_p$ and the sampling frequency $f_s$. Thus, the amount of frequencies that can be estimated and their granularity depends on the prediction horizon. Generally, the longer the prediction horizon, the more frequencies can be determined. Furthermore, a larger prediction horizon results in lower frequencies that can be determined. The frequencies themselves depend on the number of frequencies (and thus on the prediction horizon), as well as the sampling rate. Thus, the faster the MPC runs, the higher the frequencies that can be determined. The further we look into the future, the lower the frequencies we can fit in. So we need an adequate Hp. Also it means that frequency control at higher frequencies is more effective than of low frequencies.

For each motion channel $m$ in the set of motion channels $m \in \mathbb{M} = \{\omega_x, \omega_y, \omega_z, f_x, f_y, f_z\}$ we define:

$$\boldsymbol{m} = \begin{bmatrix} m_0 & m_1 & \ldots & m_{H_p-1} \end{bmatrix}^\top \tag{6}$$

with the specific forces $\boldsymbol{f}$ and the rotational rates $\boldsymbol{\omega}$, which allows us to write the frequency domain values $\mathcal{M}$ of each motion channel as:

$$\mathcal{M} = \boldsymbol{W}\boldsymbol{m}. \tag{7}$$

The novel frequency-domain tracking (FDT) term added to the cost function (1a) is:

$$J_f = w_f \sum_{m \in \mathbb{M}} w_m (\mathcal{M}_S - \mathcal{M}_V)^H \boldsymbol{F} (\mathcal{M}_S - \mathcal{M}_V), \tag{8}$$

which equals:

$$J_f = w_f \sum_{m \in \mathbb{M}} w_m (\boldsymbol{m}_S - \boldsymbol{m}_V)^\top \boldsymbol{W}^H \boldsymbol{F} \boldsymbol{W} (\boldsymbol{m}_S - \boldsymbol{m}_V) \tag{9}$$

with $\boldsymbol{F} = \text{diag}(f_0, f_1, ..., f_{H_p-1})$ and where $(\cdot)^H$ is the Hermitian transpose. This formulation allows us to select three types of weights:

$w_f$ : relative importance assigned to the added frequency cost term. It allows to trade-off frequency vs. time domain considerations;
$\boldsymbol{F}$ : relative importance assigned to different frequency components in the tracking error signal. The importance of each frequency component can be taken from models of human perception, which, underlining the importance of motion frequency, themselves are also frequency-based (Telban, Cardullo, and Kelly, 2005).
$w_m$ : relative importance of the each motion channel $m$.

The resulting cost function term is therefore:

$$J = w_t \underbrace{\sum_{n=0}^{H_p-1} (\boldsymbol{y}_{S,n} - \boldsymbol{y}_{V,n})^\top \boldsymbol{Q}_t (\boldsymbol{y}_{S,n} - \boldsymbol{y}_{V,n}) + \boldsymbol{u}_n^\top \boldsymbol{R} \boldsymbol{u}_n}_{\text{time-domain tracking}} + w_f \underbrace{\sum_{m \in \mathbb{M}} w_m (\boldsymbol{m}_S - \boldsymbol{m}_V)^\top \boldsymbol{Q}_f (\boldsymbol{m}_S - \boldsymbol{m}_V)}_{\text{frequency-domain tracking (FDT)}}. \quad (10)$$

with $\boldsymbol{Q}_f = \boldsymbol{W}^H \boldsymbol{F} \boldsymbol{W}$.

This cost function thus aims to minimize the difference in Fourier transforms of the vehicle and simulator motion. Alternatively, it is possible to construct a cost function that penalizes or prioritizes certain frequency bands in the simulator motion.

## 2.2. Quadratic Program Formulation

### 2.2.1. State Space Formulation

To represent the simulator's motion system, a state space system is used to represent a single motion channel:

$$x_{n+1} = \boldsymbol{A} x_n + \boldsymbol{B} u_n, \quad (11)$$

$$y_n = \boldsymbol{C} x_n + \boldsymbol{D} u_n, \quad (12)$$

with:

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ \boldsymbol{B} &= \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \\ 1 \end{bmatrix}, \\ \boldsymbol{C} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \\ \boldsymbol{D} &= 0. \end{aligned} \quad (13)$$

for a typical translational acceleration system with position, velocity, and acceleration as states, i.e., $x = [p,\ v,\ a]^\top$.

The present state space system only provides the calculation of the state at the next time step $n+1$. As in any typical MPC, the future states up to the prediction horizon $H_p$ must be estimated to find the best solution. This is done by the recursive application of the system (Ellensohn, 2020). For example, for the third time step, the system output equals:

$$y_{S,3} = \boldsymbol{C}(\boldsymbol{A} x_2 + \boldsymbol{B} u_2), \quad (14)$$

which can be further expanded as:

$$\begin{aligned} y_{S,3} &= \boldsymbol{C}(\boldsymbol{A}(\boldsymbol{A} x_1 + \boldsymbol{B} u_1) + \boldsymbol{B} u_2) \\ &= \boldsymbol{C}(\boldsymbol{A}(\boldsymbol{A}(\boldsymbol{A} x_0 + \boldsymbol{B} u_0) + \boldsymbol{B} u_1) + \boldsymbol{B} u_2) \end{aligned} \quad (15)$$

The recursive relationships at the points $\boldsymbol{y}_S = [y_{S,0},\ y_{S,1},\ y_{S,2},\ y_{S,3},\ \ldots, y_{S,H_p}]$, describing the simulator motion, can be captured by the matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$, such that:

$$\boldsymbol{y}_S = \boldsymbol{\Phi} x_0 + \boldsymbol{\Gamma} \boldsymbol{u} \quad (16)$$

where:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{C}\boldsymbol{A} \\ \boldsymbol{C}\boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{H_p} \end{bmatrix}, \quad (17)$$

and:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{C}\boldsymbol{B} & 0 & 0 & \cdots & 0 \\ \boldsymbol{C}\boldsymbol{A}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{B} & 0 & \cdots & 0 \\ \boldsymbol{C}\boldsymbol{A}^2\boldsymbol{B} & \boldsymbol{C}\boldsymbol{A}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{B} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}\boldsymbol{A}^{H_p-1}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{A}^{H_p-2}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{A}^{H_p-3}\boldsymbol{B} & \cdots & \boldsymbol{C}\boldsymbol{B} \end{bmatrix}. \quad (18)$$

### 2.2.2. Multiple DoF structure

For a system with multiple Degrees of Freedom (DoFs) and reference signals to track, the total state space representation can be obtained by using block diagonal matrices of each individual DoF (Bruschetta, Cenedese, and Beghi, 2019). In a similar manner, the matrix $W$ for a multi-DoF system can be constructed using a block diagonal matrix:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_{f_x} & 0 & \ldots & 0 \\ 0 & \boldsymbol{W}_{f_y} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \boldsymbol{W}_{\omega_z} \end{bmatrix} \quad (19)$$

and analogously for the frequency weighting matrix $\boldsymbol{F}$.

## 2.3. Cost Function Formulation

The general purpose of the MPC is then to find the optimal sequence of inputs $\boldsymbol{u}$ to bring the sequence of outputs $\boldsymbol{y}_S$ as close as possible to the reference signal.

Rewriting (10), using the recursive relation of (16):

$$\begin{aligned} J(\boldsymbol{u}) = {} & (\boldsymbol{\Phi} x_0 + \boldsymbol{\Gamma}\boldsymbol{u} - \boldsymbol{y}_V)^H (w_t \boldsymbol{Q}_t + w_f \boldsymbol{Q}_f)(\boldsymbol{\Phi} x_0 + \\ & \boldsymbol{\Gamma}\boldsymbol{u} - \boldsymbol{y}_V) + \boldsymbol{u}^H \boldsymbol{R} \boldsymbol{u}. \\ = {} & \boldsymbol{u}^H \boldsymbol{\Gamma}^H \boldsymbol{Q} \boldsymbol{\Gamma} \boldsymbol{u} + x_0^H \boldsymbol{\Phi}^H \boldsymbol{Q} \boldsymbol{\Gamma} \boldsymbol{u} + \boldsymbol{u}^H \boldsymbol{\Gamma}^H \boldsymbol{Q} \boldsymbol{\Phi} x_0 - \\ & \boldsymbol{u}^H \boldsymbol{\Gamma}^H \boldsymbol{Q} \boldsymbol{y}_V - \boldsymbol{y}_V^H \boldsymbol{Q} \boldsymbol{\Gamma} \boldsymbol{u} + \boldsymbol{u}^H \boldsymbol{R} \boldsymbol{u} + \\ & x_0^H \boldsymbol{\Phi}^H \boldsymbol{Q} \boldsymbol{\Phi} x_0 - x_0^H \boldsymbol{\Phi}^H \boldsymbol{Q} \boldsymbol{y}_V - \boldsymbol{y}_V^H \boldsymbol{Q} \boldsymbol{\Phi} x_0 + \\ & \boldsymbol{y}_V^H \boldsymbol{Q} \boldsymbol{y}_V, \end{aligned} \quad (20)$$

with $\boldsymbol{Q} = w_t \boldsymbol{Q}_t + w_f \boldsymbol{Q}_f$ for shorter notation. To then solve the optimization problem, quadratic solvers typically require the cost function to be of the form:

$$J(\boldsymbol{u}) = \tfrac{1}{2} \boldsymbol{u}^\top \boldsymbol{H} \boldsymbol{u} + \boldsymbol{f}^\top \boldsymbol{u}. \quad (21)$$

with $\boldsymbol{H}$ the Hessian and $\boldsymbol{f}$ the gradient vector. Note that the first term, involving the Hessian $\boldsymbol{H}$, depends quadratically on $\boldsymbol{u}$. When comparing with (20), only the first term is of quadratic form, such that:

$$\boldsymbol{H} = 2(\boldsymbol{\Gamma}^H(w_t\boldsymbol{Q}_t + w_f\boldsymbol{Q}_f)\boldsymbol{\Gamma} + \boldsymbol{R}) \tag{22}$$

The second to fifth terms of (20) depend linearly on $\boldsymbol{u}$, as does $\boldsymbol{f}$ in (24). Thus we find:

$$\boldsymbol{f} = 2\boldsymbol{\Gamma}^H(w_t\boldsymbol{Q}_t + w_f\boldsymbol{Q}_f)(\boldsymbol{\Phi}x_0 - \boldsymbol{y}_V) \tag{23}$$

The remaining terms in (20) do not depend on $\boldsymbol{u}$ at all, meaning that when optimizing for $\boldsymbol{u}$ (the function of the MPC), these terms are unaffected and can thus be neglected from the cost function.

The specific challenge when incorporating frequency control is that quadratic solvers can only handle a completely real cost function. This is made clear by the use of the transpose operator $(\cdot)^\top$ in (24), suitable only for real numbers, rather than the Hermitian transpose $(\cdot)^H$.

Considering the present case of complex-valued components, the matrix $\boldsymbol{H}$ and vector $\boldsymbol{f}$, which contain both real and imaginary parts due to the frequency components, must be rewritten. This can be achieved by splitting the real and imaginary components, and passing both as separate, but real components to the cost function, i.e.,:

$$J(\boldsymbol{u}_r) = \tfrac{1}{2}\boldsymbol{u}_r^\top \boldsymbol{H}_r\boldsymbol{u}_r + \boldsymbol{f}_r^\top \boldsymbol{u}_r \tag{24}$$

Starting with $\boldsymbol{H}$:

$$\boldsymbol{H}_{real} = \text{Re}\left(\boldsymbol{\Gamma}^H[w_t\boldsymbol{Q}_t + w_f\boldsymbol{Q}_f]\boldsymbol{\Gamma} + \boldsymbol{R}\right), \tag{25}$$

and

$$\begin{aligned}\boldsymbol{H}_{imag} &= \text{Im}\left(\boldsymbol{\Gamma}^H[w_t\boldsymbol{Q}_t + w_f\boldsymbol{Q}_f]\boldsymbol{\Gamma} + \boldsymbol{R}\right]) \\ &= \text{Im}\left(\boldsymbol{\Gamma}^H w_f\boldsymbol{Q}_f\boldsymbol{\Gamma}\right]),\end{aligned} \tag{26}$$

as $\boldsymbol{Q}_t$ and $\boldsymbol{R}$ are real-valued. The Hessian matrix $\boldsymbol{H}$, which is complex valued, can now be represented by mapping from complex to real space, constructing a real-valued matrix of its real and imaginary components (Stickan, et al., 2019), such that $\boldsymbol{H} \in \mathbb{C}^{H_p\times H_p} \rightarrow \mathbb{R}^{2H_p\times 2H_p}$:

$$\boldsymbol{H}_r = \begin{bmatrix}\boldsymbol{H}_{real} & -\boldsymbol{H}_{imag} \\ \boldsymbol{H}_{imag} & \boldsymbol{H}_{real}\end{bmatrix}. \tag{27}$$

Thus, instead of supplying the complex-valued $\boldsymbol{H}$ to the cost function, the matrix in (27) can be used. Similarly for $\boldsymbol{f}$:

$$\boldsymbol{f}_{real} = \text{Re}\left(2\boldsymbol{\Gamma}^H(w_t\boldsymbol{Q}_t + w_f\boldsymbol{Q}_f)(\boldsymbol{\Phi}x_0 - \boldsymbol{y}_V)\right), \tag{28}$$

and:

$$\begin{aligned}\boldsymbol{f}_{imag} &= \text{Im}\left(2\boldsymbol{\Gamma}^H(w_t\boldsymbol{Q}_t + w_f\boldsymbol{Q}_f)(\boldsymbol{\Phi}x_0 - \boldsymbol{y}_V)\right) \\ &= \text{Im}\left(2\boldsymbol{\Gamma}^H(w_f\boldsymbol{Q}_f)(\boldsymbol{\Phi}x_0 - \boldsymbol{y}_V)\right),\end{aligned} \tag{29}$$

Similarly as for $\boldsymbol{H}$, $\boldsymbol{f}$ can be expressed by:

$$\boldsymbol{f}_r = \begin{bmatrix}\boldsymbol{f}_{real} \\ \boldsymbol{f}_{imag}\end{bmatrix}. \tag{30}$$

Because of the mapping $\boldsymbol{H}$ and $\boldsymbol{f}$ to real space, the input $\boldsymbol{u}$ must be mapped to real space as well.

$$\boldsymbol{u}_r = \begin{bmatrix}\boldsymbol{u}_{real} \\ \boldsymbol{u}_{imag}\end{bmatrix} \tag{31}$$

As we only consider real, time-domain inputs, we can write:

$$\begin{bmatrix}\boldsymbol{u} \\ \boldsymbol{0}_{H_p\times 1}\end{bmatrix} \tag{32}$$

Thus, the zero-valued $\boldsymbol{u}_{imag}$ is only added to ensure compatibility with $\boldsymbol{H}$ and $\boldsymbol{f}$.

# 3. Results

This section analyzes the performance of the frequency-aware MPC MCA. To this end, the proposed algorithm is compared against a benchmark MPC MCA that is not explicitly frequency-aware, i.e., does not contain the FDT in the cost function, but is identical otherwise. First a multisine signal is used as a reference for a simplified analysis. Afterwards, the longitudinal motion of a reference vehicle drive is used to extend the analysis.

## 3.1. Multisine Simulation Example

First, as an example, simulations of the presented control problem are performed on a synthetic sum of sinusoids (multisine) single DoF reference signal:

$$y_V = 5\cdot\cos(\omega_1 t + \frac{\pi}{2}) + 0.2\cdot\cos(\omega_{21} t + \frac{\pi}{4}). \tag{33}$$

Multisines are ideal for investigating the frequency behavior of a system (Kolff, et al., 2019; Kolff, et al., 2023) as they contain discrete frequencies. However, in the present case of the frequency control, the frequencies of the multisines must match the frequencies that the DFT can estimate. The reference signal therefore contained two discrete frequencies: $\omega_1 = 6.28319$ rad/s and $\omega_{10} = 62.83$ rad/s, which represent the first and tenth harmonics of the DFT with $H_p = 250$ s and $\Delta t = 0.004$ s, representing one relatively low and high frequency respectively. To simplify the control task, the simulations were performed offline using a recorded drive. The optimizations were performed using Quadprog in Matlab R2025a. In the control problem, the input was limited to $2\,\text{m/s}^2$.

The result of the MPC simulation, without the FDT (i.e., $w_f = 0$), is shown by the red line in Figure 1. As this approach minimizes the absolute difference between the reference and simulator signals only, this leads to the situation where the MPC employs a constant $2\,\text{m/s}^2$ output to remain as close as possible to the reference signal, therefore ignoring the high-frequency content of the reference signal. This is likely to be detrimental for realism in driving simulation, as the complete high-frequency content beyond $2\,\text{m/s}^2$ is not produced.

The green line in Figure 1 shows the MPC *with* FDT. The weighting to the frequency cost function term was set to $w_f = 10$ to activate and prioritize the frequency-domain tracking. The time-domain tracking weight was set to zero. The two frequency components were weighted differently ($F_1 = 10$, $F_{10} = 10,000$), to represent a situation in which high-frequency is strongly prioritized for in the simulation. With this FDT, the MPC strategy is different. As a result, the controller deliberately provides a less strong low-frequency motion, but at the crucial benefit of providing the full high-frequency motion of $\omega_{10}$, fitting exactly within the acceleration limits of $2\,\text{m/s}^2$.

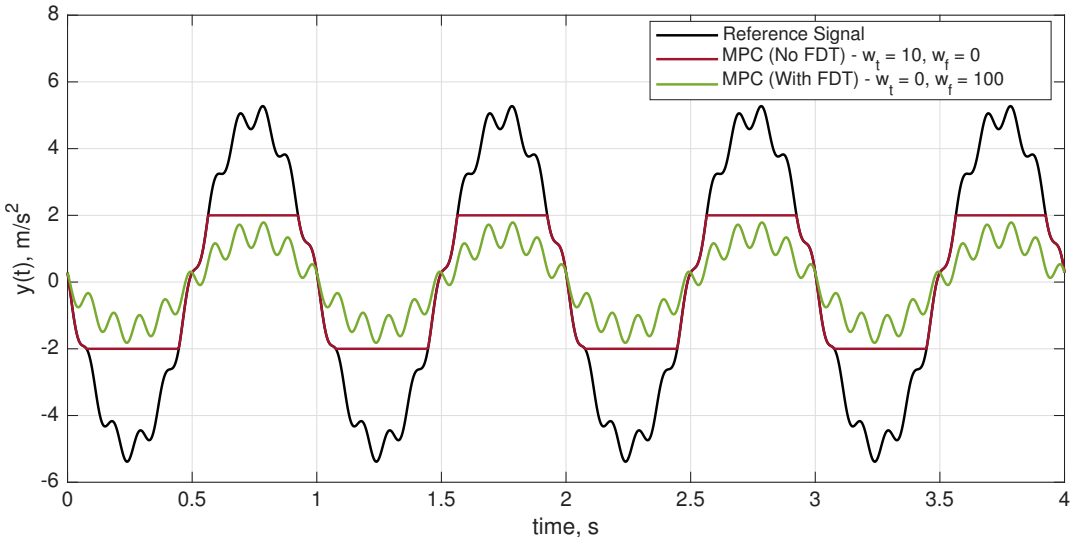



Figure 1: Outputs of the MPC algorithms with and without FDT, highlighting the effectiveness in prioritizing high-frequency oscillations in the case with FDT control for a multisine reference signal.

This allows for the complete reproduction of the high-frequency motion, in contrast to the standard MPC without DFT.

Figure 2 shows the Bode diagram of the three signals, with the magnitude in the top subfigure. Note that the MPC without FDT, due to its partially constant output, produces a uniform frequency spectrum, strongly differing from the discrete frequencies in the reference signal. This is a result of the non-multisine output signal of this MPC without FDT. In contrast, the MPC with FDT indeed only produces the motion of the two frequencies $\omega_1$ and $\omega_{10}$. Furthermore, the lower subfigure highlights the ability of the MPC with FDT to also reproduce the multisines with the correct phases; the phases of the input signal of $\pi/2$ $(90°)$ and $\pi/4$ $(45°)$, respectively corresponding to the frequencies $\omega_1$ and $\omega_{10}$.

## 3.2. Vehicle Drive Evaluation

As a second evaluation of the method, a recording of a realistic rural vehicle drive is used for the tracking in two DoFs. Here, the reference signals are the longitudinal $(f_x)$ and lateral $(f_y)$ specific force recorded in a previous driving simulation experiment in a rural driving scenario, lasting $80\,\mathrm{s}$.

Similar to the multisine use-case, the two methods (with and without FDT) are compared. In this case, the MPC without FDT had weights $w_t = 10$ and $w_f = 0$. For the MPC with FDT, the best results were found using a combination of time-domain and frequency-domain tracking, i.e., $w_t = 2.5$ and $w_f = 100$. In the present use-case, again a prioritization on the high-frequency components is made. For this reason, the applied frequency weights were of the linearly increasing form $F_k = 5k/2N$, where $F_k$ represents the diagonal entries of the weighting matrix $\boldsymbol{F}$. This weighting function is shown in Figure 3.

The resulting output signals are shown in Figures 4a (longitudinal specific force) and 4b (lateral specific force), together with the reference signals. Correctly, the MPC with FDT remains below the limits to avoid a distortion in the frequency content, unlike the MPC

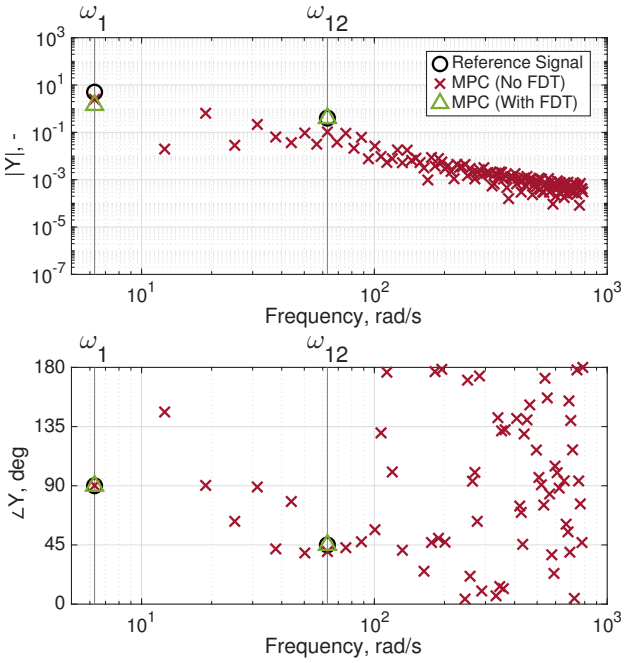



Figure 2: Bode diagram of the multisine reference signal and the MPC methods with and without FDT at $t = 0$.

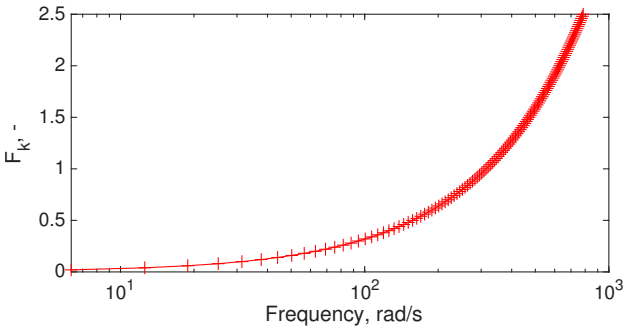
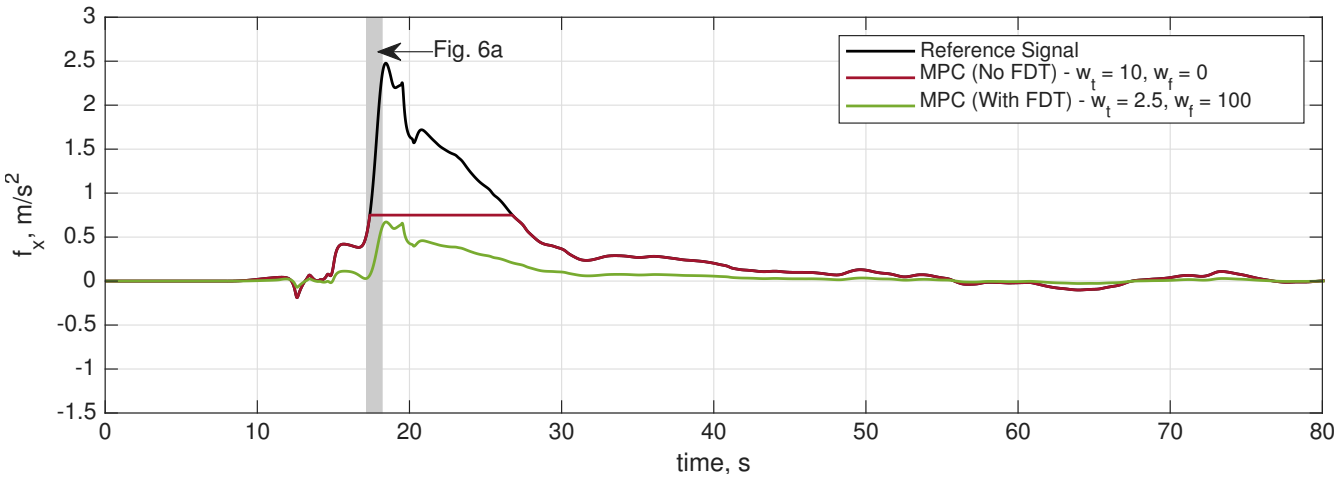
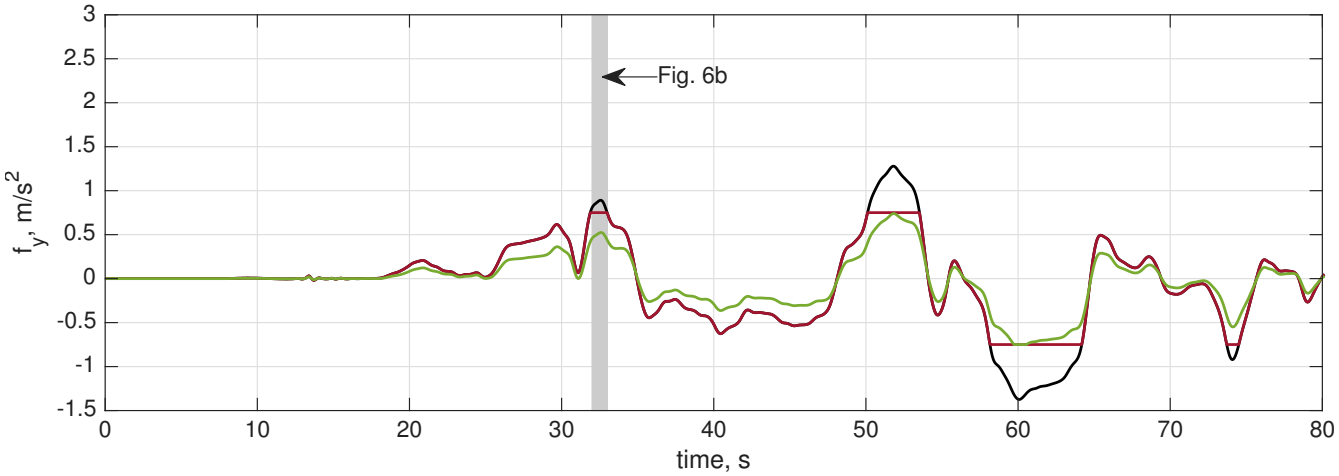



Figure 3: Weighting function representing the diagonal entries of the matrix $\boldsymbol{F}$.




(a) Longitudinal specific force ($f_x$)




(b) Lateral specific force ($f_y$)

Figure 4: Outputs of the MPC algorithms with and without FDT, highlighting the effectiveness in prioritizing high-frequency oscillations in the case with FDT control for a realistic use-case. The grey areas represent “snap-shots” of a single optimization step, which are further analyzed in the Bode diagrams in Figure 5.

without FDT. Overall, the motion is scaled down. This may be due to the additional input required by the cost function to justify a frequency-domain deviation, while scaling the motion does not impair the frequency-domain consistency between the reference and MPC output. Alternatively, an additional cost due to a frequency deviation may arise from moving “back” closer to the reference, e.g., at around $30\,\mathrm{s}$.

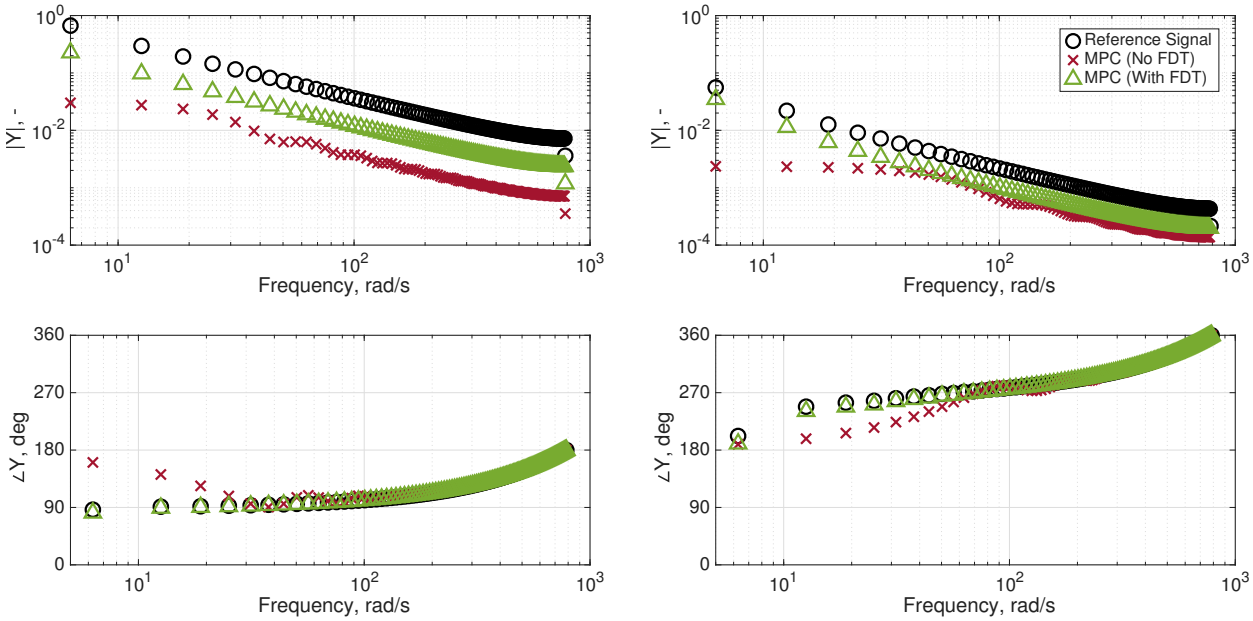
The grey areas represent a snapshot of the optimization lasting the prediction horizon $H_p$. These points of interests are further investigated through their Bode diagrams to compare their frequency-domain behavior. These are shown in Figures 5a ($f_x$) and 5b ($f_y$). From both the magnitude and phase responses, it is clear that the MPC with FDT indeed provides a frequency response highly consistent with the reference signal, whereas the MPC without FDT does not.

# 4. Discussion

This paper introduces a frequency-domain approach for MPC in driving simulation. This is achieved by transforming the predicted output of the simulator motion to the frequency-domain using a DFT matrix. The frequency cost term is then included alongside the usual time-domain tracking cost, resulting in a hybrid formulation that retains the standard predictive structure. With this additional frequency control in the MPC cost function, the controller can selectively suppress or amplify specific frequency components in the simulator motion. The presented MPC application can therefore greatly improve the frequency-domain characteristics of the motion, which has been one of the main drawbacks of existing MPC approaches in driving simulation.

## 4.1. Implications

Including frequency tracking in the MPC formulation imply that both the frequency content of the motion cueing can be actively steered by controlling both the magnitude and the phase of the MPC output. This may have the benefit of a better control in periodic motions, such as road curvature, lane-keeping oscillations, steering corrections, or vibrations. How the components in the frequency domain are handled depends on the implementation of the DFT in the cost




(a) Longitudinal specific force ($f_x$)

(b) Lateral specific force ($f_y$)

Figure 5: Bode diagrams of the signals in the vehicle drive evaluation.

function. The presented implementation, where specific frequency bands are prioritized, is only one type of application. This can, for example, be used to focus on and match those frequencies that are most important for the human perception. Similarly, certain frequency bands in the MPC output can be actively avoided, e.g., to adhere to system constraints. This cannot be guaranteed when using an approach in which these frequency ranges are filtered beforehand in the reference signal, as a non-FDT MPC could then still produce motion in these frequency ranges. With FDT, these frequency bands can be actively avoided, liberating workspace for other frequency ranges, and providing a key benefit over filter-based approaches.

As another type of FDT implementation, rather than minimizing the differences for specific frequency ranges, it is possible to optimize for providing consistency among the whole frequency range. This would then optimize for making the difference between vehicle and simulator motion as similar as possible for all discrete frequencies. These different implementations will require a different structure of the FDT in the MPC cost function. It is also recommended to investigate which weighting functions of the matrix $\boldsymbol{F}$ are best suited for these different applications, rather than the trial-and-error approach that was used for the simulations used in the present paper.

## 4.2. Limitations

While DFT can thus improve the frequency-domain characteristics of the MPC output, there are also drawbacks to the approach. First, one primary concern is the discretization inherent in the DFT. Because the frequency resolution is determined by the prediction horizon length, the controller can only penalize content at specific discrete frequencies. This poses a challenge if undesirable content appears at discrete frequencies between the frequencies of the DFT, which would not be effectively penalized. A possible solution would be to let the discrete frequencies represent frequency “bins” (Kolff, et al., 2019), in which a whole range of frequencies around the discrete frequencies are included and evaluated in the cost function.

A second limitation comes from the possible application of the DFT method in real-time MPC. In time-domain MPC techniques, the sample time $\Delta t$ and the prediction horizon $H_p$ are commonly based on the available computational power. As both the sample time and the prediction horizon of the MPC determine the frequencies of the DFT, obtaining the DFT at specific frequencies restricts the values of both the sample time and the prediction horizon. The prediction horizon must for example be sufficiently long to be able to estimate low enough frequencies. The real-time performance of an MPC may also limit the sample time $\Delta t$, and the prediction horizon $H_p$, which in return affect the frequencies of the DFT. A restricted computational power may thus lead to ineffective frequency-domain tracking. For these reasons, the inclusion of frequency tracking in MPC may only be recommended if there is a specific reason to do so.

Third, the inclusion of the frequency-domain tracking inherently increases the computational effort to solve the control problem. The transformation of the frequency-domain cost into a real-valued quadratic form, by splitting the frequency-domain terms into real and imaginary parts, is a key step in making the approach feasible. Nevertheless, this increases the size of the $\boldsymbol{H}$ and $\boldsymbol{f}$ matrices, leading to an increase in computational effort. For real-time control applications, this may limit the number of frequencies that can be effectively included, requiring a careful selection of the most relevant spectral components to penalize.

## 4.3. Future Work

As a direction for future work, the frequency control can be implemented on the input level as well. In the current approach, the DFT MPC optimizes the inputs in the time-domain (real inputs) to control the output in the frequency-domain (complex values). Instead, it is also possible to provide frequency-domain inputs by allowing the input vector $\boldsymbol{u}$ to include imaginary values as well, i.e., by making the inputs complex. This may be especially useful if (part of) the reference signal is known to be periodic. A benefit of this approach can be a reduced computational load, as the MPC only needs to optimize for the most suitable frequency-domain characteristics, rather than long time series.

Another clear next step in future work is the evaluation of the motion by using human test drivers to investigate whether the case with FDT is preferred. Furthermore, not only the perceptual fidelity is of interest. Humans exhibit band-limited control, meaning that they do not respond to all disturbances equally across the frequency spectrum. Furthermore, the ability of humans to adapt to situations is known to depend on the frequency characteristics of the in- and output signals (McRuer and Jex, 1967). Adding frequency control in the cost function allows the MPC to mirror these frequency-based characteristics, which may thus be a crucial step to enhance *behavioral* fidelity in MPC as well.

Investigating the control behavior would necessarily imply testing the FDT method in real-time applications. Future work will need to investigate how the prediction quality of the real-time simulation may affect the frequency control. It is further possible that different frequency components of the prediction may be predicted with different certainties. For example, engine vibrations are likely strongly linked to the road type and type of engine.

# 5. Conclusion

This work introduces a frequency-domain cost function formulation for MPC that allows for control of the simulator output motion in the frequency domain. By penalizing selected frequency components using the DFT matrix method, the approach offers explicit control over the frequency contents of the signal, while remaining close to the standard MPC formulation. The simulations results confirm that the method can effectively suppress undesired frequencies while maintaining accurate tracking, both for the multisine reference as well as the realistic driving use-case. Although future work is required to investigate the application in closed-loop driving simulations with human participants, the presented methods provide a fundamental basis for frequency control in MPC, such as to optimize the motion for human perception or to explicitly include the platform bandwidth limitations in the optimization. Therefore, the presented methodology provides a promising foundation for frequency-based MPC, including applications in flight simulation and more general human-centered control scenarios.

# References


Beghi, A., Bruschetta, M., and Maran, F., Dec. 2012. A Real Time Implementation of MPC Based Motion Cueing Strategy for Driving Simulators. In: *Proceedings of the 51st IEEE Conference on Decision and Control*. Maui, HI, USA, pp. 6340–6345.

Biemelt, P., Böhm, S., Gausemeier, S., and Trächtler, A., 2021. Subjective Evaluation of Filter- and Optimization-Based Motion Cueing Algorithms for a Hybrid Kinematics Driving Simulator. In: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Melbourne, Australia, pp. 1619–1626.

Bruschetta, M., Maran, F., and Beghi, A., 2017a. A Fast Implementation of MPC-based Motion Cueing Algorithms for Mid-Size Road Vehicle Motion Simulators. *Vehicle System Dynamics*, 55(6), pp. 802–826.

Bruschetta, M., Cenedese, C., and Beghi, A., 2019. A Real-Time, MPC-based Motion Cueing Algorithm with Look-Ahead and Driver Characterization. *Transportation Research Part F: Traffic Psychology and Behaviour.*, 61, pp. 38–52.

Bruschetta, M., Maran, F., and Beghi, A., Mar. 2017b. A Nonlinear, MPC-Based Motion Cueing Algorithm for a High-Performance, Nine-DOF Dynamic Simulator Platform. *IEEE Transactions on Control Systems Technology*, 25(2), pp. 686–694.

Cleij, D., Venrooij, J., Pretto, P., Katliar, M., Bülthoff, H. H., Steffen, D., Hoffmeyer, F., and Schöner, H.-P., 2019. Comparison between filter- and optimization-based motion cueing algorithms for driving simulation. *Transportation Research Part F: Traffic Psychology and Behaviour*, 61, pp. 53 –68.

Dagdelen, M., Reymond, G., Kemeny, A., Bordier, M., and Maïzi, N., 2009. Model-based predictive motion cueing strategy for vehicle driving simulators. *Control Engineering Practice*, 17(9), pp. 995–1003.

Ellensohn, F., Hristakiev, D., Schwienbacher, M., Venrooij, J., and Rixen, D., 2019. Evaluation of an Optimization Based Motion Cueing Algorithm Suitable for Online Application. In: *Proceedings of the Driving Simulation Conference 2019 Europe*. Strasbourg, France, pp. 93–100.

Ellensohn, F., 2020. Urban Motion Cueing Algorithms - Trajectory Optimization for Driving Simulators. PhD Dissertation. Technische Universität München.

Fang, Z., Wautier, D., and Kemeny, A., 2022. FFT based optimal MCA for AD/ADAS driving tests. In: *Proceedings of the Driving Simulation Conference 2022 Europe*. Strasbourg, France, pp. 119–126.

Freund, R. M., 2004. *Solution Methods for Quadratic Optimization*. Tech. rep. Massachusetts Institute of Technology.

Grandia, R., Farshidian, F., Dosovitskiy, A., Ranftl, R., and Hutter, M., 2019. Frequency-Aware Model Predictive Control. *IEEE Robotics and Automation Letters*, 4(2), pp. 1517–1524.

Kolff, M., Van der El, K., Pool, D. M., Van Paassen, M. M., and Mulder, M., 2019. Approximating Road Geometry with Multisine Signals for Driver Identification. In: *Fourteenth IFAC Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, September 16-19*. Tallinn, Estonia, pp. 341–346.

Kolff, M., Venrooij, J., Schwienbacher, M., Pool, D. M., and Mulder, M., 2023. The Importance of Kinematic Configurations for Motion Control of Driving Simulators. In: *IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. Bilbao, Spain, pp. 1000–1006.

McRuer, D. T. and Jex, H. R., 1967. A Review of Quasi-Linear Pilot Models. *IEEE Transactions on Human Factors in Electronics*, HFE-8(3), pp. 231–249.

Qazani, M. R. C., Asadi, H., and Nahavandi, S., 2020. A New Gantry-Tau-Based Mechanism Using Spherical Wrist and Model Predictive Control-Based Motion Cueing Algorithm. *Robotica*, 38(8), pp. 1359–1380.

Stickan, B., Rutquist, P., Geyer, T., and Diehl, M., 2019. Spectrum Shaping Methods for Predictive Control Approaches Applied to a Grid-Connected Power Electronics Converter. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5223–5230.

Telban, R. J., Cardullo, F. M., and Kelly, L. C., 2005. *Motion Cueing Algorithm Development: New Motion Cueing Program Implementation and Tuning*. Tech. rep. NASA/CR-2005-213746. State University of New York.

Wang, Y. and Boyd, S., Mar. 2010. Fast Model Predictive Control Using Online Optimization. *IEEE Transactions on Control Systems Technology*, 18(2), pp. 267–278.