Motion Cueing Quality Comparison of Driving Simulators using Oracle Motion Cueing

Kolff, M.J.C.; Venrooij, Joost ; Schwienbacher, Markus; Pool, D.M.; Mulder, Max

Publication date
2022

Document Version
Final published version

Published in
Driving Simulation Conference Europe 2022

Citation (APA)
Kolff, M. J. C., Venrooij, J., Schwienbacher, M., Pool, D. M., & Mulder, M. (2022). Motion Cueing Quality Comparison of Driving Simulators using Oracle Motion Cueing. In *Driving Simulation Conference Europe 2022*

Motion Cueing Quality Comparison of Driving Simulators using Oracle Motion Cueing

Maurice Kolff[1,2], Joost Venrooij[1], Markus Schwienbacher[1], Daan M. Pool[2] and Max Mulder[2]

(1) BMW Group, Research, New Technologies, Innovations, 80788 Munich, e-mail : {Maurice.Kolff, Joost.Venrooij, Markus.Schwienbacher}@bmw.de
(2) Delft University of Technology, Faculty of Aerospace Engineering, section of Control & Simulation, 2629 HS Delft, e-mail : {m.j.c.kolff, d.m.pool, m.mulder}@tudelft.nl

Abstract - *BMW's new driving simulation center operates multiple motion-base simulators – each with a different kinematic configuration – to serve various experiment use-cases and requirements of simulator users. The selection of a simulator for each experiment should ideally be based on their relative strengths and weaknesses. To support this decision-making process, subjective and objective predictions of motion cueing quality can be used. This paper provides an example comparison of four motion-base driving simulators. The kinematic configurations of the simulators considered differed in the additional presence of a yaw-drive and/or a linear xy-drive. The comparison is made by calculating offline, optimization-based motion cueing with perfect prediction capabilities (the 'Oracle') for nine urban drives. A prediction of subjective motion incongruence ratings is made for each simulator. In addition, an error type identification method is used (identifying scaling, missing cue, false cue and false direction cue errors) and evaluated per simulator. As Oracle can fully utilize the available workspace, the employed evaluation methods provide an insight in the fundamental capabilities of each simulator. Both the modelled ratings and the error type analysis show the benefits of adding a xy-drive in urban use-cases: predicted ratings reduce by 19% (i.e., better), while scaling and missing cue errors in the yaw rate are reduced when adding a yaw-drive. The presence of both of these additional motion systems allow for practically one-to-one and therefore error-free motion cueing. The proposed methods provide a straight-forward, yet insightful basis for simulator selection. The presented methods can be extended towards the analysis of multiple motion cueing algorithms and/or other use-cases for systematically selecting the best-suited motion cueing method.*

Keywords: *Motion cueing, simulator comparison, quality comparison, objective assessment.*

1. Introduction

BMW has recently constructed a novel integrative driving simulation center, hosting multiple driving simulators with different kinematic configurations. The centre is aimed at covering a wide scope of driving simulation use-cases. The availability of these simulators opens up a new design problem, where for any upcoming experiment or use-case to be performed on a simulator, the best-suited method (motion cueing algorithm and simulator) must be selected. In order to make informed choices in this selection process, a reliable assessment of motion cueing quality is needed. Due to the large scope of available motion cueing methods, an assessment method preferably should allow for an "off-site" analysis as much as possible, to avoid having to test each motion cueing method on all available simulators.

Assessment methods of motion cueing methods can be based on objective or subjective analyses. The latter refer to evaluations of the motion cueing given by a driver and are only obtainable through on-site testing in a simulator. Objective evaluations, on the other hand, can be obtained without a driver. Typically, these are metrics of motion cueing quality based on the motion set-points of the simulator with respect to the reference motion. Such evaluations can be performed without the actual simulator and are therefore faster to obtain and use. This allows for a more systematic assessment of the available motion cueing methods. The main difficulty in finding and using objective metrics is that it is still not fully understood what entails good or bad motion cueing (Casas-Yrurzum, et al., 2020). So far, no accepted objective metrics have been found that can fully replace subjective evaluations.

In literature, recent work by Cleij, et al. (2018) helps by providing a bridge between both assessment types. In their presented method, subjects continuously rate the perceived motion cueing through a rating interface while passively being driven around, such that a continuous rating signal is obtained. First attempts by Ellensohn, et al. (2019b) and, more recently, Kolff, et al. (2022) aimed at investigating the relationship between the subjective ratings as given by the participants and objective metrics. A different method is to objectively analyze the error types that occur in the motion cueing by the *type* of motion cueing mismatch. Examples of this are the analysis of false cues (Ariel and Sivan, 1984; Pham and Nguyen, 2021; Salisbury and Limebeer, 2014) or missing cues (Cleij, 2020). Here, it is generally accepted that false cues are more detrimental than a missing cue, whereas an error due to scaled-down motion is the least detrimental. Both predictions of subjective ratings and error type evaluation methods may be used to select the best-suited simulator and motion cueing algorithm (MCA), based on their motion cueing quality.

The goal of this paper is to compare four motion-base simulators with different kinematic configurations using objective methods. For each simulator, an offline optimization-based algorithm is used to calculate the motion cueing output for nine reference drives in an urban scenario. Based on subjective motion cueing quality ratings, obtained from Kolff, et al. (2021), a developed model from Kolff, et al. (2022) is used to predict the continuous motion incongruence ratings for the four simulators. Furthermore, an algorithm for error type classification is presented and evaluated for all four simulators as a purely objective analysis. These analyses facilitate the understanding the fundamental differences per motion system configuration. Secondly, it helps to answer the question which simulator is best suited for urban use-cases.

The structure of this paper is as follows: Section 2 describes the background of the problem. Section 3 presents the simulators used for analysis and describes the evaluation methods, where section 4 presents its results. Points of discussion and further steps are discussed in section 5. The paper is concluded in section 6.

2. Problem Statement

2.1. Motion Cueing Trinity

Amongst users of motion-based driving simulators, the proper and use-case-specific implementation and testing of motion cueing is often considered to be one of the most difficult tasks. Each considered motion cueing solution can be characterized in a motion cueing trinity (Figure 1), constituted of three elements:

1. the use-case, which can refer to the type of scenario (such as urban, highway or rural), as well as the type of experiment, such as the evaluation of driving dynamics or human-machine-interaction studies
2. the used motion cueing algorithm with its corresponding configuration, such as the tuning parameters
3. the simulator's motion system on which the motion cues are generated.

The best possible motion cueing solution is only obtained when the best combination of the three elements of the trinity is selected. Such a selection process typically starts with one or two of these elements fixed. Note that this selection process is a multi-dimensional design problem, in which all three elements are related. For example, the benefits of a more difficult to implement MCA may only be worthwhile for a particular combination of simulator and use-case.

In all three building blocks of this trinity, recent developments extend the scope of options. In an attempt to bring an increasing amount of on-road testing to the simulated world, the amount of use-cases to be handled in simulation has increased significantly. Secondly, the development of new MCAs, such as Model-Predictive Control (MPC) algorithms (Dagdelen, et al., 2009) enables moving away from the conservative and difficult to tune Classical Washout Algorithm (CWA). MPC often provides better cueing due to its optimization-based strategy (Ellensohn, et al., 2019a), although its added benefit in motion cueing quality might not always be justified, considering the additional complexity.

The availability of various driving simulators increases the complexity of the design problem. It is possible that, although MPC is capable of providing better motion cueing, its advantages might be more or less pronounced on certain kinematic configurations. Therefore, it is not necessarily true that a one-size-fits-all solution for the available driving simulators exists. To make well-informed decisions, being able to better *predict* and systematically compare how the various motion cueing solutions perform relative to each other is therefore necessary.

2.2. Predictions of Motion Cueing Quality

An additional difficulty is that in the design stages of an experiment – in which choices regarding the motion cueing solution have to be made – the actual solution cannot be experienced in the simulator yet. In this phase, little details of the experiment are known, although the designer has to make an informed estimate on which combination would work out well. For this reason, it is useful to develop measures that can accurately predict the motion cueing quality as it would be perceived by participants.

Cleij, et al. (2018) proposed a rating method, in which participants give *continuous* motion incongruence ratings (MIRs) based on the perceived motion incongruences (PMIs). A motion incongruence is defined as a perceived deviation from the expected (real) vehicle motion. These MIRs are therefore indications of how participants think the motion they perceive matches that of the real vehicle. Participants are driven through an environment passively, rather than driving themselves, so that they can fully focus on the rating method. The rating is measured on a scale from 0 to 10 and can be changed at any point in time, resulting in a continuous rating signal. A value of 0 indicates no mismatch at all, whereas the highest score of 10 indicates the worst motion cueing encountered in the experiment.

This subjective rating method has been implemented in various studies to compare different types of algorithms (Cleij, et al., 2019; Cleij, et al., 2018; Ellensohn, et al., 2019a; Ellensohn, et al., 2020; 2019b;

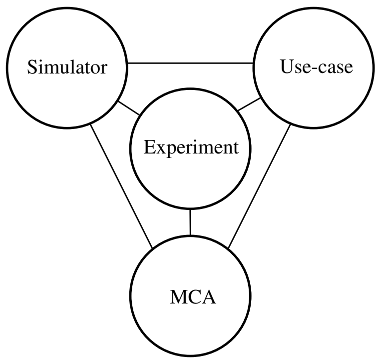

Figure 1: Motion cueing trinity, indicating the multi-dimensional design problem.

Van der Ploeg, et al., 2020). Recently, this method was used in a realistic urban scenario (Kolff, et al., 2021). The recorded continuous rating signals, having a high temporal resolution, have been used in Kolff, et al. (2022) to acquire a time-domain model, relating predicted average ratings as function of the errors in the motion cueing that are presented in the simulator. Such predictions allow for an analysis of incongruences, overall or in specific sections of interest, to evaluate the strengths and weaknesses of the motion cueing.

Extending MIR predictions by an analysis of the various types of errors that occur in the motion cueing, fundamental strengths and weaknesses of the motion systems can be explicitly revealed. Therefore, in the following sections, an analysis is performed in which four state-of-the-art simulators are compared with the same motion cueing algorithm (use-case fixed, MCA fixed, simulator variable).

3. Methods

3.1. Simulators

For our comparison, four simulators are selected, each with different motion systems:

- The *Vega Vector (VV)* (Cruden B.V., 2021, Figure 2a) is the smallest simulator under investigation and consists of a 6-DoF hexapod (Stewart platform) with an actuator stroke of 64 cm. Its main novel feature is a cylindrical $220°$ LED-wall that allows for high contrast visuals combined with high brightness and vivid colors.
- The *Sirius Vector (SV)* (Cruden B.V., 2021, Figure 2b) has the same hexapod as the Vega Vector, but has an additional $\pm 175°$ yaw-drive underneath, resulting in a 7-DoF system. The yaw-drive is especially useful for reproducing large yaw motion ranges. The LED-wall is similar to the Vega Vector, but covers the full $360°$ horizontal field-of-view.
- The *Ruby Space (RS)* (VI-Grade, 2015, Figure 2c) is a 9-DoF system (hexapod on tripod) that has been used in a number of recent evaluation studies (Ellensohn, et al., 2019a; Ellensohn, et al., 2020; 2019b; Kolff, et al., 2021). This simulator was used for the acquisition of the continuous rating data for the model development by Kolff, et al. (2022). Its tripod can rotate $\pm 25°$, but also has a 1.5 m workspace in both x and y-directions.
- The *Sapphire Space (SS)* (Bosch Rexroth B.V. (Since May 2022: Van Halteren Technologies B.V.) and AVSimulation, 2021, Figure 2d) is BMW's largest simulator (9-DoF). Its XY-drive allows movement over an area of 19.14 m $\times$ 15.7 m. Furthermore, it includes a large 6-DoF hexapod (total stroke of 1.15 m) with a single DoF $\pm 175°$ yaw-drive on top. The visuals are projected by a full $360°$ projection system inside the enclosed dome.

3.2. Input Data Collection

Driving data are collected in a human-in-the-loop experiment, where participants drive through an urban environment in the Sapphire Space simulator. From this experiment, nine drives are recorded. The driven route is the same as in Kolff, et al. (2021), although with one additional corner and traffic light at the end of the scenario. The driven route is identical for all participants, but due to differences in the driving style (such as velocity and position in the lane), all drives are inherently different. The acquired signals of the nine drives therefore cannot be compared in the time-domain, as the time at which certain maneuvers are encountered differs per driver. For this reason, all analyses are performed as a function of the along-track distance:

$$a(t) = \sum_{i}^{N} \sqrt{\Delta x^2 + \Delta y^2}, \tag{1}$$

where Δx_i^2 and Δy_i^2 are the incremental distance per unit time in the global coordinate system. Theoretically, even the along-track distance does not enable a fully fair comparison, as maneuvers can be spaced at different points in the along-track distance signals per drive due to differences in lane position. However, from visual inspection of the data, this effect may be considered negligible.

3.3. Oracle Motion Cueing Algorithm

A comparison between the four simulators is made using a non-real-time optimization-based cueing strategy with infinite prediction horizon, here referred to as *Oracle*. This MCA calculates the simulator's motion that minimizes the cost function:

$$J = \sum_{i=1}^{N} \tfrac{1}{2}(y-\hat{y})^T Q (y-\hat{y}) + \sum_{i=0}^{N-1} \tfrac{1}{2} x^T S x, \tag{2}$$

where $y = [f_x,\, f_y,\, f_z,\, \omega_x,\, \omega_y,\, \omega_z]^T$ are the true measured specific forces and rotational rates at the driver position to be reproduced by the algorithm. $\hat{y}$ is the actual output of the algorithm, with $Q = \text{diag}(Q_f,\, Q_\omega)$ the diagonal weight matrix for this reference tracking term. The second term is a state excitation penalization of the states x, pulling the simulator back to its neutral state. Although, in principle, a washout mechanism is not required for Oracle motion cueing, it is still added to increase the convexity to the optimization (Katliar, 2020). The state term consists of the states of the various motion subsystems. Its corresponding weighting matrix is a diagonal matrix $S \in \mathbb{R}^{n \times n}$ with $n = 3\times$DoF, as it contains weights for the position, velocity and acceleration signals of each degree-of-freedom. The optimization furthermore used the system limits on position and acceleration level as hard constraints. The algorithm contains the full non-linear kinematic descriptions of the analyzed simulators and is based on the implementation of Ellensohn, et al. (2019b). The optimization problems were programmed using CasADI (Andersson, et al., 2018) and optimized with IPOPT (Wächter and Biegler, 2006).

The optimization is performed along all samples $N = \frac{t}{dt}$ of each run, with t the duration of the run and dt the sampling time, set to 10 ms. Oracle can calculate, under these pre-defined weight settings, the optimal simulator motion by minimizing the cost function in Equation (2) for a set of reference driving data. As the solution depends on the applied weights, we do not claim that such optimization-based cueing can be considered perfect and even the word optimal must be used with caution. Furthermore, an optimal solution of Oracle is not necessarily synonymous with the best possible subjective motion cueing quality.

(a) Vega Vector.

(b) Sirius Vector.

(c) Ruby Space.

(d) Sapphire Space.

Figure 2: The four selected simulators for comparison. Pictures are property of BMW Group.

The Oracle algorithm and its output motion can only be used in simulations where participants are driven through the environment rather than driving themselves, as for real-time driving the future states are inherently unknown. Due to the full exploitation of future states, Oracle serves as a theoretical limit of what simulators can achieve. The algorithm is therefore specifically suited to investigating system differences, because it can fully exploit the available workspace. To properly use Oracle as the quality benchmark of a simulator, the corresponding weights were tuned for each simulator, as shown in Table 1, which were selected by a trial-and-error approach. Note that the output weights for the rotational rates were set ten times higher as the specific forces, as various sources (Drop, et al., 2018; Katliar, 2020; Van der Ploeg, et al., 2020) have shown this to result in a good balance. The output weight of the yaw rate deviates and was set to 100, to emphasize the tracking in this channel.

In total 36 Oracle outputs are generated: four simulator outputs for each of the nine reference drives. An example of the outputs for one drive is given in Figure 3(a-c). Note that visually the outputs for all simulators match reasonably well with the reference signals. The computational time per run differs per simulator structure, ranging from ten minutes for the Vega Vector up to one hour for the Sapphire Space.

3.4. Rating Prediction Model

To evaluate the motion cueing quality, a prediction model of continuous motion incongruences was used. Based on the previously performed offline rating experiment using continuous ratings on the Ruby Space simulator (Kolff, et al., 2022), system identification of the rating data has shown that the rating behaviour of humans in urban scenarios can be described by the delayed, low-pass filtered and weighted combination of the longitudinal and lateral specific force mismatches. Therefore, first the total weighted error contribution is calculated:

$$E(t) = W_{f_x}(|f_x^v - f_x^s|) + W_{f_y}(|f_y^v - f_y^s|). \quad (3)$$

The weights W_{f_x} and W_{f_y} weight the relative contributions and are set to 1.17 and 1.63, respectively. The predicted rating is calculated by simulating the total weighted error $E(t)$ with the transfer function:

$$H_{x,y}(j\omega) = \frac{\omega_{c_{x,y}}}{j\omega + \omega_{c_{x,y}}} e^{-j\omega\tau}, \quad (4)$$

where $\tau = 0.008$ s is the time delay constant and $\omega_{c,y} = 0.37$ rad/s and $\omega_{c,x} = 0.26$ rad/s are the cut-off frequencies for the y and x channels, respectively. These parameters have been shown to well describe the average measured rating signals of 50 participants (Kolff, et al., 2022).

As mentioned in section 3.2, the rating signals are calculated for each of the nine runs, but cannot be

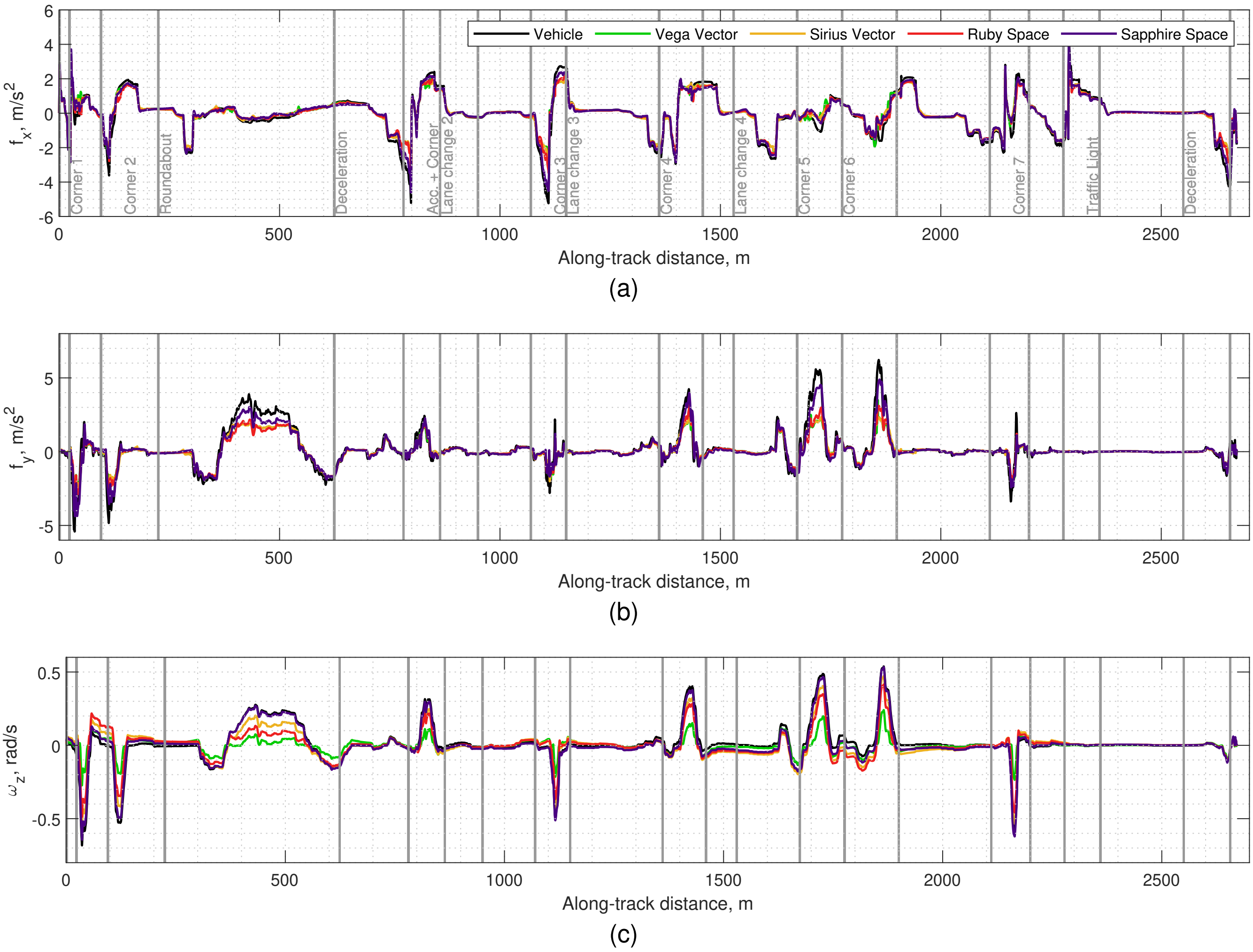

Figure 3: Oracle outputs of the four simulators, compared to the vehicle data (input), for data set 2.

compared in the time domain directly. Therefore, the time-domain ratings are converted to a function of the along-track distance $a(t)$, i.e., the ratings are interpolated to an equidistant along-track distance.

3.5. Error Classification Algorithm

Supplementary to the rating prediction model, an analysis is made of the different error types that can occur due to the motion cueing. Four different error types are considered: false direction cues, false cues, missing cues, and scaling errors. A graphical example of the interpretation of these types is shown in Figure 4. The classifications are calculated for each motion signal $m \in \{f_x, f_y, f_z, \omega_x, \omega_y, \omega_z\}$ and are defined as:

- False direction cue error (4): Both simulator and vehicle are moving above the threshold of perception p (such that $m_s > p$ and $m_v > p$), but in opposite directions. This is considered the worst error possible.
- False cue error (3): The simulator is moving ($m_s > p$), where the vehicle is not ($m_v \leq p$).
- Missing cue error (2): The simulator produces no motion ($m_s \leq p$), whereas the vehicle is ($m_v > p$).
- Scaling error (1): The simulator moves ($m_s > p$) in the same direction as the vehicle ($m_v > p$), but either weakened or too strong. Because humans

Table 1: Output- and state weights used in the Oracle optimization, per simulator. Single values indicate identical values in each direction.

	Output weights		State weights									
	$Q_{f_x f_y f_z}$	$Q_{\omega_x \omega_y \omega_z}$	$S_{\phi\theta\psi}$	$S_{\dot\phi\dot\theta\dot\psi}$	S_{xyz}	$S_{\dot x\dot y\dot z}$	$S_{\ddot x\ddot y\ddot z}$	S_{ψ_t}	$S_{\dot\psi_t}$	$S_{x_t y_t}$	$S_{\dot x_t \dot y_t}$	$S_{\ddot x_t \ddot y_t}$
Vega Vector	1	[10, 10, 100]	10	10	1	1	1	-	-	-	-	-
Sirius Vector	1	[10, 10, 100]	10	10	1	1	1	1	10	-	-	-
Ruby Space	1	[10, 10, 100]	10	10	1	1	1	1	10	0.04	0.04	0.04
Sapphire Space	1	[10, 10, 100]	10	10	1	1	1	1	10	0.01	0.01	0.01

can have a range of acceptable scaling factors that they still consider as coherent, the scaling errors are only defined if the simulator is moving at less than 70% or more than 130% of the vehicle motion (Van Leeuwen, et al., 2019).

- Correct cue (0): If none of the other error types are detected, the cue is considered to be correct.

Note that the motion signals m_s and m_v refer to specific forces and rotational rates of the simulator and vehicle, respectively. The error types are calculated as signals (i.e., over time) for each of these signals separately, resulting in six separate classifications. Only one error type can be possible at each time step. The error type perception thresholds are $p = 0.05$ m/s^2 for the specific forces (Reymond and Kemeny, 2000) and $p = 3$ deg/s for the rotational rates (Reid and Nahon, 1985).

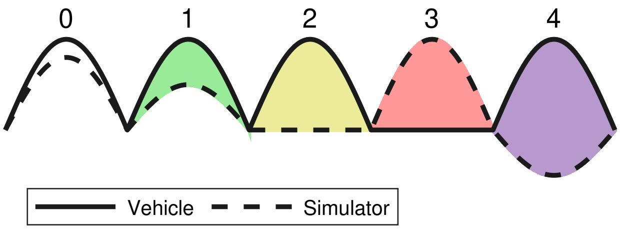

Figure 4: Example of the error types between vehicle and simulator motion: No error (0), scaling error (1), missing cue error (2), false cue error (3) and false direction cue error (4).

3.5.1. Error Area per Unit Time

To compare the contribution of the error types, the error area per unit time (EAT) is calculated. For example, for a Type 1 (scaling) error in the specific force channel f_x, the EAT is defined as:

$$EAT_{1,f_x} = \frac{\int_0^T |\epsilon_1(t)(f_x^v(t) - f_x^s(t))|dt}{T}, \quad (5)$$

where T is the total time of the drive. The signal $\epsilon_1(t)$ is a boolean signal that is either zero when there is no error of such type, or one when that error type is present. The EAT is calculated for each of the six reference signals and for each of the five error type separately, leading to 30 EATs. The EAT represents how much of the reference signal is reproduced incorrectly through the error type. It can be zero, in the case where no of such errors are present in the signal. It is unbounded on the upper side, as there is no single case that defines a 'complete error'.

4. Results

4.1. Rating Prediction Model

First, the rating prediction model is used to estimate the motion incongruence ratings for each simulator. Figure 5 shows the predicted ratings as a function of along-track distance a, where the curves indicate the average across the nine drives. The spread indicates one standard deviation from the mean. Note that lower ratings indicate higher quality (Cleij, et al., 2018). For each simulator, the root-mean square of the rating signal is calculated. The Sapphire Space overall performs best throughout the runs, with an overall root-mean square of $RMS_{SS} = 0.50$, while for the Ruby Space this is $RMS_{RS} = 0.61$. Note that also when considering Figure 5, the differences in RMS represent the differences in motion cueing quality over time: Sapphire Space induces the lowest ratings for all maneuvers, whereas Ruby Space is second-best. The Vega Vector and Sirius Vector perform the worst ($RMS_{VV} = 0.68$ and $RMS_{SV} = 0.69$), and produce nearly identical ratings. This can be explained by considering the yaw-rate error is not included in the rating model, such that the Sirius Vector shows no clear benefit in the calculated ratings.

4.2. Error Type Classifications

Figure 6 shows the distribution of the four error types in terms of the EAT. In contrast with the predicted ratings, the vertical specific force f_z, roll rate ω_x and pitch rate ω_y are included in the analysis as well, to investigate the effect on these channels (such as the dependency on tilt-coordination). When regarding the longitudinal specific force f_x, the four simulators (denoted in the figure along the x-axis) show no clear differences, indicating that these cues can be reproduced well by a simple hexapod system. Furthermore, there are no differences in the vertical specific force f_z, as the route was completely flat.

On the other hand, a difference between simulators is visible in the reproduction of the lateral specific force f_y. Although the Vega Vector, Sirius Vector, and Ruby Space perform equally well in this channel, the Ruby Space can do so while decreasing the false cues in ω_y by a factor of 2, indicating a reduced dependency on tilt-coordination. By far the best performing simulator in these two coupled channels is the Sapphire Space, as its extensive motion capabilities of the xy-drive allow for reducing the use of tilt-coordination and therefore avoiding *any* false cue in ω_y (and similarly for ω_x). This shows that even though differences in the output of the motion cueing as presented in Figures 3(a-c) might seem small, there is a profound difference in how the motion cueing is spread-out over the capabilities of the various motion systems.

The largest differences are visible in the yaw rate. The Vega Vector (Denoted in the figure with VV), clearly lacks the yawing ability as it has an average EAT of 0.18 of the runs classified as a scaling error, 0.046 for a missing cue, followed by an EAT of 0.001 for false cues. The presence of the yaw-drive for Sirius Vector (SV) decreases both scaling and missing cue errors, i.e., it reproduces more yaw motion, but at the cost of slightly more false cues. By far the best performing simulator is the Sapphire Space (SS), with a low EAT in the false cue category; the scaling, missing, and false direction cue errors are zero.

5. Discussion

The results in this paper show that with 'Oracle' motion cueing, fundamental differences between the motion systems of different motion-base driving simulators can be quantitatively compared. In the presented results, the main kinematic differences between the four simulators directly result in different cueing capabilities. Although not included in the rating model, the largest differences in error types were

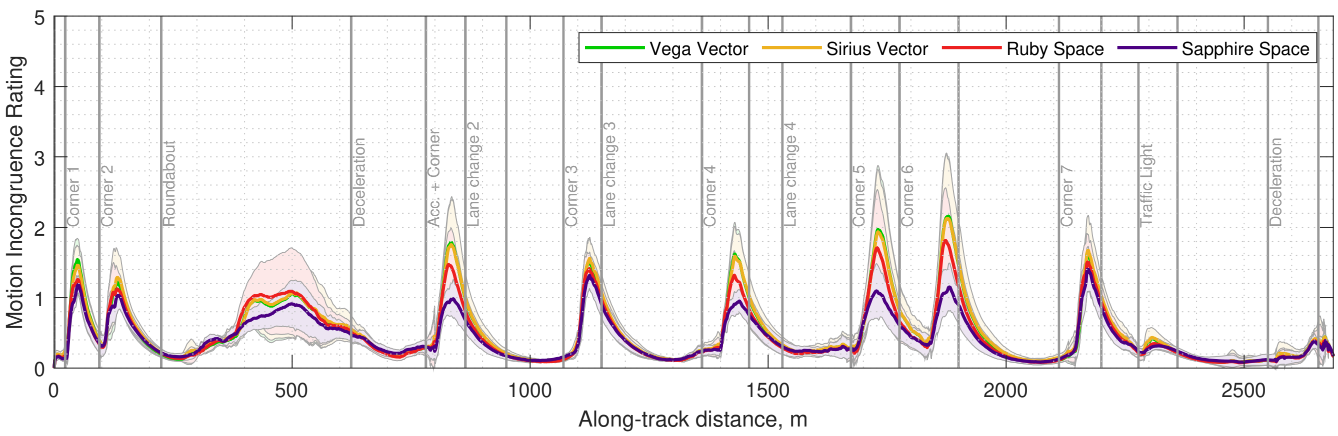

Figure 5: Predicted motion incongruence ratings with standard deviations, calculated per simulator.

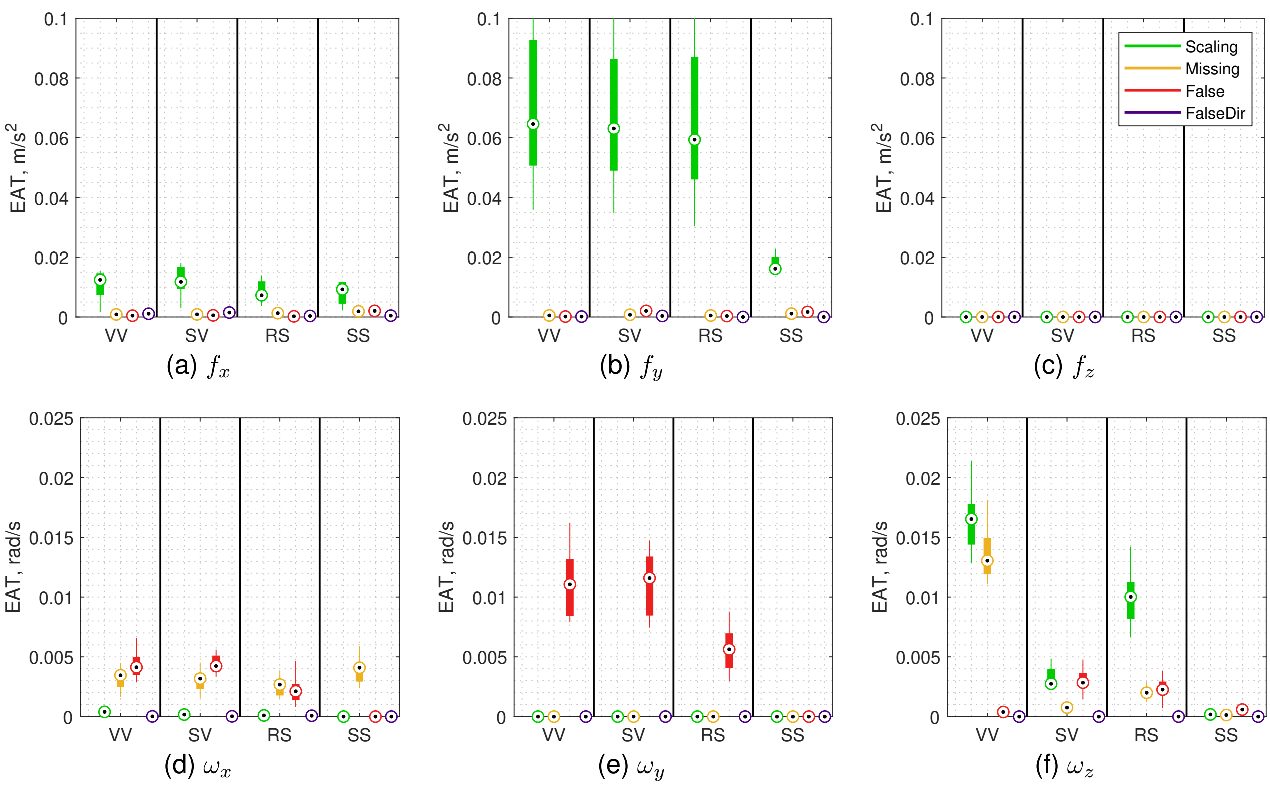

Figure 6: Distributions of the error types, per simulator. Horizontal bars indicate the sum of the average error types of each simulator. VV = Vega Vector, SV = Sirius Vector, RS = Ruby Space and SS = Sapphire Space.

found in the yaw rates. The amount of scaling and missing cue errors in the motion cueing are strongly reduced, as it is difficult to produce yaw motion using only a hexapod (Vega Vector). To reduce the errors in the yaw rate channel, a yaw-drive therefore has a direct benefit for the urban use-case in question. The Ruby Space does provide less false cues in the roll- and pitch rate channels, but due to its limited tripod rotation still has relatively large scaling errors in the yaw rate. It could therefore be more suitable for a use-case that requires fast translational motion, such as lane changing, but less yaw motion. The benefit of a (large) xy-drive is nevertheless clear, with the Sapphire Space performing best in the ratings and all channels of the error detection algorithm.

In the selection of a suitable simulator (and MCA) for the presented use-case, it must be evaluated which quality indicators are trusted. As we have presented two different methods, the natural question arises as to *which* method should be used, as both have various (dis)advantages. The modelled ratings provide an intuitive prediction of cueing quality. It is easy to rank motion cueing solutions (for example using the RMS), while at the same time providing information of the cueing quality over time. Arguably, the weakness of this approach is that the model will only be sensitive to errors in the channels on which the model was constructed (i.e., the lateral and longitudinal specific force error, in this case). An extended and better validated model specific for a given use-case or scenario would require collecting more subjective rating data.

The error classifications better show the distribution of different cueing errors in the various channels. However, the method itself provides little information regarding the relative (subjective) importance of these channels. It is therefore possible that an objective analysis of motion cueing quality requires multiple tools that supplement each other. In fact, the presented use-case shows an example of this: the continuous ratings show a ranking of the simulators, while the error classifications further help understanding how these ratings are produced between the different simulators and how the rating model can be potentially improved.

To apply the presented methods in a selection process for the full motion cueing trinity (see Section 2), different MCAs would have to be evaluated as well. Although decreasing scaling and missing cue errors can be achieved without increasing the amount of false cues for Oracle, there is no guarantee that this will be true for other MCA types, because an increase in magnitude (i.e., less scaling) could also increase the magnitude of false cues. Furthermore, we expect that the amount of false direction cue errors increases when using an non-optimization-based algorithm. An example would be a classical washout algorithm, where motion in the opposite direction is common due to the linear washout and therefore generating a cue in the false direction. Future work could therefore focus on extending the presented methods by including various MCAs in the analysis, thereby adding a degree-of-freedom to the design problem of the motion cueing trinity framework, as indicated in Figure 1. Only then can the best-suited motion cueing solution for a full array of options be selected.

6. Conclusion

In this paper, an objective comparison between four motion-base driving simulators is presented. A prediction of subjective ratings is made, as well as an objective assessment through a motion cueing error type classification (separating scaling errors, missing cues, false cues and false direction cue errors). The simulator movement is calculated for each simulator using an 'Oracle' algorithm (optimization-based with infinite prediction horizon). The results show that both the addition of a yaw-drive and xy-drive decrease the false and missing cues in the related channels. The presence of an xy-drive increases the predicted subjective motion cueing quality. The combination of predictions of motion incongruence ratings and the error classification algorithm can form the basis of a combined objective-subjective assessment method of simulators, motion cueing algorithms, and use-cases, such that the best-suited motion cueing solution can be selected.

References

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M., 2018. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*.

Ariel, D. and Sivan, R., 1984. False Cue Reduction in Moving Flight Simulators. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14(4), pp. 665–671. `https://doi.org/10.1109/ICSMC.1990.142165`.

Casas-Yrurzum, S., Portalés-Ricart, C., Morillo-Tena, P., and Cruz-Neira, C., 2020. On the Objective Evaluation of Motion Cueing in Vehicle Simulations. *IEEE Transactions on Intelligent Transportation Systems*, 99, pp. 1–13. `https://doi.org/10.1109/TITS.2020.2978498`.

Cleij, D., Feb. 2020. Measuring, modelling and minimizing perceived motion incongruence. PhD Dissertation. Delft University of Technology.

Cleij, D., Venrooij, J., Pretto, P., Katliar, M., Bülthoff, 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. `https://doi.org/10.1016/j.trf.2017.04.005`.

Cleij, D., Venrooij, J., Pretto, P., Pool, D. M., Mulder, M., and Bülthoff, H. H., Feb. 2018. Continuous Subjective Rating of Perceived Motion Incongruence During Driving Simulation. *IEEE Transactions on Human-Machine Systems*, 48(1), pp. 17–29. `https://doi.org/10.1109/THMS.2017.2717884`.

Dagdelen, M., Reymond, G., Kemeny, A., Bordier, M., and Maïzi, N., Apr. 2009. Model-based predictive motion cueing strategy for vehicle driving simulators. *Control Engineering Practice*, 17(2009), pp. 995–1003. `https://doi.org/10.1016/j.conengprac.2009.03.002`.

Drop, F. M., Olivari, M., Katliar, M., and Bülthoff, H. H., 2018. Model Predictive Motion Cueing: Online Prediction and Washout Tuning. In: *Proceedings of the Driving Simulation Conference 2018 Europe, Antibes*.

Ellensohn, F., Hristakiev, D., Schwienbacher, M., Venrooij, J., and Rixen, D., 2019a. Evaluation of an Optimization Based Motion Cueing Algorithm Suitable for Online Application. In: *Proceedings of the Driving Simulation Conference 2019 Europe, Strasbourg, France*, pp. 1–8.

Ellensohn, F., Spannagl, M., Agabekov, S., Venrooij, J., Schwienbacher, M., and Rixen, D., 2020. A hybrid motion cueing algorithm. *Control Engineering Practice*, 97, p. 104342.

Ellensohn, F., Venrooij, J., Schwienbacher, M., and Rixen, D., 2019b. Experimental evaluation of an optimization-based motion cueing algorithm. *Transportation Research Part F*, 62, pp. 115–125. `https://doi.org/10.1016/j.trf.2018.12.004`.

Katliar, M., Dec. 2020. Optimal control of motion simulators. PhD Dissertation. Albert-Ludwigs-Universität Freiburg.

Kolff, M., Venrooij, J., Schwienbacher, M., Pool, D. M., and Mulder, M., 2021. Quality Comparison of Motion Cueing Algorithms for Urban Driving Simulations. In: *Proceedings of the Driving Simulation Conference 2021 Europe VR*. Munich, Germany.

— 2022. Identification of Continuous Subjective Evaluations of Motion Cueing in Urban Driving Simulations. *Submitted*.

Pham, D.-A. and Nguyen, D.-T., 2021. False cue influence on motion cue quality for 10 motion cueing algorithms. *Science Progress*, 104(3), p. 00368504211036857. `https://doi.org/10.1177/00368504211036857`.

Reid, L. and Nahon, M., 1985. *Flight Simulation Motion-Base Drive Algorithms: Part 1. Developing and Testing the Equations*. Tech. rep. Technical Report UTIAS 296. University of Toronto, Institute for Aerospace Studies.

Reymond, G. and Kemeny, A., 2000. Motion Cueing in the Renault Driving Simulator. *Vehicle System Dynamics*, 34(4), pp. 249–259. `https://doi.org/10.1076/vesd.34.4.249.2059`.

Salisbury, I. G. and Limebeer, D. J. N., 2014. An Algorithm for False Cue Reduction and Prepositioning. In: *Proceedings of the Driving Simulation Conference 2014 Europe, Paris, France*.

Van der Ploeg, J. R., Cleij, D., Pool, D. M., Mulder, M., and Bülthoff, H. H., 2020. Sensitivity Analysis of an MPC-based Motion Cueing Algorithm for a Curve Driving Scenario. In: *Proceedings of the Driving Simulation Conference 2020 Europe, Antibes, France*.

Van Leeuwen, T. D., Cleij, D., Pool, D. M., Mulder, M., and Bülthoff, H. H., Feb. 2019. Time-varying perceived motion mismatch due to motion scaling in curve driving simulation. *Transportation Research Part F: Traffic Psychology and Behaviour*, 61, pp. 84–92. `https://doi.org/10.1016/j.trf.2018.05.022`.

Wächter, A and Biegler, L., 2006. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), pp. 25–57.