sciendo

LATVIAN JOURNAL OF PHYSICS
AND TECHNICAL SCIENCES
2024, N 4

DOI: 10.2478/lpts-2024-0024

IMPLEMENTATION OF CLASSICAL MOTION CUEING ALGORITHMS FOR SERIAL MANIPULATOR BASED MOTION SIMULATOR: LARRMOS CASE STUDY

A. Stupans*, P. Maksimkins, A. Senfelds, L. Ribickis

Riga Technical University,
Institute of Industrial Electronics, Electrical Engineering and Energy,
12-1 Azenes Str., Riga, LV-1048, LATVIA
*e-mail: andrejs.stupans@rtu.lv

The study focuses on the development and evaluation of a motion control system for the Large Range Robotic Motion Simulator (LaRRMoS), which is based on a serial manipulator with 6 degrees-of-freedom. Two motion cueing algorithms (MCAs) – Classical Washout Filter (CWF) and Classical Washout Filter with Cylindrical coordinates (CYL) – are implemented and tested for simulation fidelity and LaRRMoS workspace utilization. CWF and CYL motion cueing performance gets assessed in two use cases: car simulation and luge sports simulation. The car case is a typical driving simulator application. The luge sports case is chosen for its high dynamics to assess the limits of the robot and motion cueing. Though this study uses a pre-recorded set of motion data and runs simulations in non-interactive mode, it aims for real-time interactive simulation for LaRRMoS in the future, so only MCAs capable of real-time operation are inspected. Conducted tests show that the CYL algorithm provides superior performance in highly dynamic motion replication, particularly in lateral accelerations.

Keywords: *Control, cueing, motion, robotics, simulation, serial manipulator.*

1. INTRODUCTION

Motion simulators have broad use nowadays. Simulators are used for pilot/driver training [1], entertainment, and research [2], [3]. According to the application, simulators have different control systems and motion platform types, they vary in fidelity level, motion capabilities, immersivity and operation costs [4], [5]. The equipment used in

simulators depends on its application. Pilot training simulators prioritize their ability to simulate special forces with great precision compared to real aircraft to better train pilots' situation awareness in highly controlled and safe environments. However, such precision may not be needed for the entertainment industry, where the motion of the simulator and the simulated vehicle is optimized for maximizing the user's positive experience and can be far from reality. Simulators used for research can be optimized for motion fidelity, immersivity or both according to the aim of the research.

Motion control system for simulators consists of the following main parts: Simulated vehicle dynamics, motion cueing algorithms (MCA), inverse kinematics (IK) solver, and motion platform controller. Safety mechanisms can be integrated into each part of the motion control system. Figure 1 shows the diagram with a general approach to motion control systems for simulators, but every implementation can have its own specific system architecture.

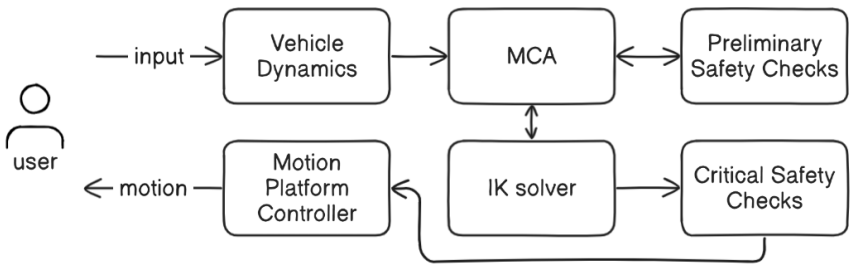

Fig. 1. General architecture of motion simulator control system.

Motion control system highly depends on the motion platform type. Stewart platform is a popular parallel-type manipulator [6]. Its main features are rigidity, precision, and dynamics [7], [8]. However, the Stewart platform is very limited in translational and rotational motion compared to serial manipulators [9]. A significantly bigger range of motion provided by serial manipulators may be more beneficial for motion simulator applications than high precision and dynamics. The hybrid-type motion platforms have the advantages of both parallel and serial manipulators, but they are of significantly increased complexity [4] and incur costs. Hybrid manipulators can have more than six degrees of freedom (DOF). Based on how much DOF a platform has, the complexity of the control system varies significantly. Advanced simulators usually have 6 DOF or more, as the motion of a body in 3D space can be fully defined with 6 DOF [4].

The aim of the research is to design a motion control system suited for a 6 DOF serial manipulator motion platform measured in terms of motion simulation fidelity and the platform workspace usage. The motion control system was tested by simulating two cases – car simulation and luge sports simulation. This study uses pre-recorded vehicle data for non-interactive simulation. The interactive simulation requires a mathematical model of vehicle dynamics that is not in the scope of the study. However, motion control system is developed to be capable of real-time closed-loop interactive simulation in the future. Motion data for both cases were recorded by a custom motion data acquisition system developed specifically to use in conjunction with the motion platform.

2. MATERIALS AND METHODS

For this study, Large Range Robotic Motion Simulator (LaRRMoS) was used as a motion platform. LaRRMoS (Fig. 2) is based on a KUKA KR600 R2830 industrial robot, equipped with a passenger cabin and all the required safety measures to carry a passenger. The control system consists of an Operator PC that runs MCA in Matlab Simulink and inverse kinematics solver in a separate process. The second PC named BEC Controller ensures that all safety systems (garage doors, motion sensors, cabin lock, and safety belt) are working properly. This functionality is provided by BEC GmbH. The Kuka KRC4 controller receives the robot position and implements the motion. Also, critical safety mechanisms are run in KRC4 via Kuka.SafeOperation. The architecture of the LaRRMoS control system used in the study is shown in Fig. 3.

Fig. 2. LaRRMoS motion platform.

The testing of the developed control system was divided into several steps. First, to generate vehicle dynamics, a body motion acquisition system was developed. Then vehicle dynamics were recorded for driving simulation and luge sport simulation. Next, two MCA algorithms were tested on LaRRMoS for both cases. The luge simulation case was chosen for extreme dynamics of motion with fast-changing orientation and high acceleration values. The car simulation was chosen as a reference to evaluate LaRRMoS performance in a typical driving simulation application. The precision of generated specific forces and the motion platform effective workspace usage were evaluated.

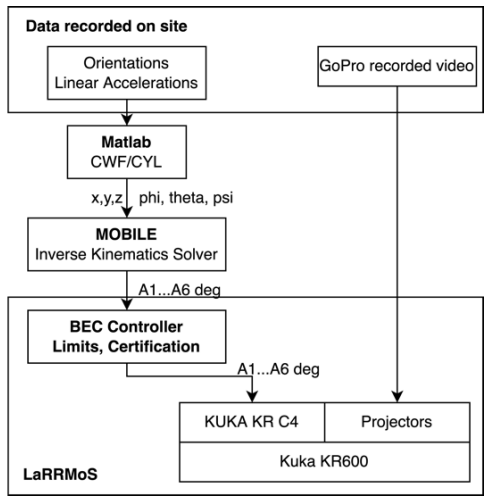

Fig. 3. LaRRMoS control system architecture used in the study.

3. VEHICLE DYNAMICS

To simulate a vehicle, its motion parameters have to be obtained first. There are multiple ways of obtaining the data. One could either develop a mathematical model of the vehicle and use the data generated by this model or obtain motion data directly from the real vehicle.

The first option can be done in game engines like Unity or Unreal using a built-in physics engine. This comes with the benefit of being integrated with advanced 3D graphics and tools for creating a virtual environment that will be used for the visual and audial part of the simulator. However, game engines are optimized for performance, and can be imprecise in physics simulations because of non-deterministic calculations as stated by [10]. The advanced simulators usually have their own mathematical model of the vehicle (e.g., DLR flight simulator by [11]) achieving high precision when simulating motion. However, this approach is a complex task on its own, developing a physics engine from scratch. There is a solution in between, using an advanced physics model as a game engine add-on, like Vehicle Physics Pro in Unity as performed in [12].

To avoid the complexity of developing and verifying the vehicle dynamic model for the time being, the second option was chosen to obtain motion data directly from a real vehicle. It goes with the disadvantage of being able to run only non-interactive simulations, where the user does not have control of a vehicle. In contrast, interactive simulation allows the user to control the simulated vehicle. To obtain vehicle motion data, it was chosen to use the custom motion parameter acquisition system proposed in [13]. This system has been developed to be a source of reliable motion data for motion cueing algorithm input.

3.1. Car Dynamics

In the case of car simulation, the physical parameters of the car should be considered. When undergoing the curve car suspension reacts to the shifting centre of mass, the car rolls to the outside of the curve, and the gravity vector generates interference on the Y-axis measured acceleration as shown in Fig. 4.

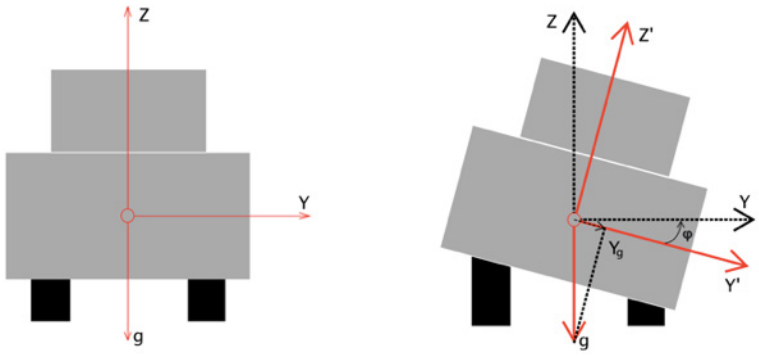

Fig. 4. Gravity vector interference during the car's turning manoeuvre [13].

The motion cueing algorithm demands the data to be free of gravity vector interference. To compensate for the gravity interference formula (1) is used:

$$\vec{a}_{compensated} = \frac{\vec{a}_{measured} - 9{,}81 * \sin(\varphi)}{\cos(\varphi)}. \quad (1)$$

To verify the proposed compensation method, the S-curve manoeuvre (also known as the reverse curve) was chosen as a test procedure shown in Fig. 5. The speed of the car was kept constant during the curve at 15 km/h. Since the velocity of the car is constant, the X-axis acceleration can be neglected. The radius of the curve was measured as 7 meters and the theoretical maximal Y-axis acceleration was calculated as $\vec{a}_{theoretical}$ using formula (2):

$$\vec{a}_{theoretical} = \frac{v^2}{R}. \quad (2)$$

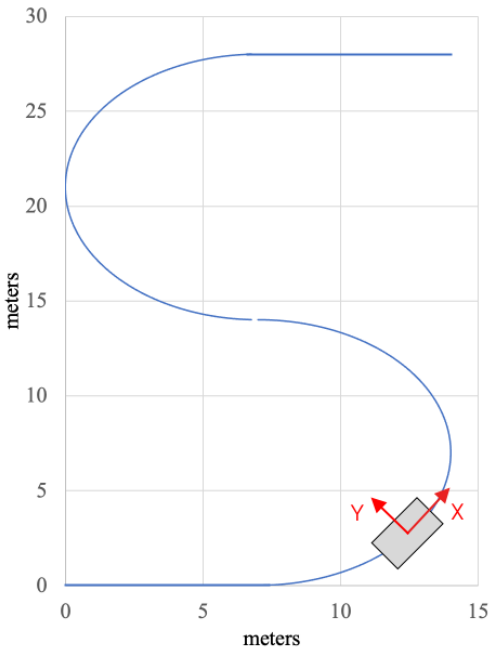

Fig. 5. S-curve manoeuvre for obtaining car dynamics [14].

3.2. Luge Dynamics

The second case is luge sports simulation. The methodology for motion data recording is kept the same as with the car simulation case. Bobsleigh track located in Sigulda, Latvia, was chosen. Its length is 1420 m and it consists of 16 curves. Athletes can reach up to 125 km/h during the track trial. Several track trials were made with the help of a skilled athlete and IMU data were recorded. GPS data was not obtained as the metal roofing of the track shielded the signal.

4. MOTION CUEING ALGORITHM

The motion cueing algorithm (MCA) transforms the simulated vehicle's motion cues into simulator motion cues. The motion of the simulator is limited in terms of workspace, speed and acceleration leading to the inability to simply replicate motions of the simulated vehicle precisely. However, it is possible to produce motion cues that are perceived by a user as real ones while operating within the motion platform limits. The method is based on features of human motion perception. A human could not distinguish a force induced by vehicle acceleration from a force induced by gravity without visual cues [15]. A human feels a specific force that is defined by formula (3). In the motion simulation context, the term specific force is often used to describe forces acting on an object excluding gravity force.

$$SF = \frac{F}{m} = a - g. \quad (3)$$

To simulate the specific force of the vehicle, the motion of a simulated vehicle is divided into three channels: Translation, Tilt Coordination and Rotation. The Translation channel simulates only high-frequency vehicle motion using short translational motion of the platform. The Tilt Coordination channel uses a slight tilt of the cabin to trick the driver's perception into feeling part of gravitational acceleration as a vehicle acceleration [16]. It produces a low-frequency part of a specific force that could not be simulated by the Translation channel. The rotation channel simulates angular speed using short rotations of the cabin. Tilt angles and rotation angles are summed up to get the final rotation of a motion platform. This approach allows to simulate specific forces of a vehicle.

The most widely used motion cueing algorithm is the Classical Washout Filter (CWF) proposed by [17]. Figure 6 shows a diagram of the CWF algorithm. The input of the filter is the simulated vehicle's linear acceleration and angular velocity ω_{IN}. Output is displacement S and tilt angles β of the motion platform end-effector. Rotation matrix ${}^{O}R_{S}$ transforms accelerations from the end-effector's reference frame to the robot base frame. Formulas for high-pass and low-pass filters are provided in [17]. Tilt coordination block transforms low-frequency accelerations into Euler angles using formula (4). Then rate of change of tilt angles is limited to prevent false rotation cues for dynamic motions. However, the limit values suggested by the [18] are for flight simulation, and this study focuses on driving and sports simulation, so the influence of the rate limiter block is also evaluated in the scope of the paper. The study [19] also states that Tilt Coordination without a rate-limiter can be acceptable for a highly dynamic Formula 1 simulator:

$$\beta_T = \begin{bmatrix} 0 & 1/g & 0 \\ -1/g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}. \tag{4}$$

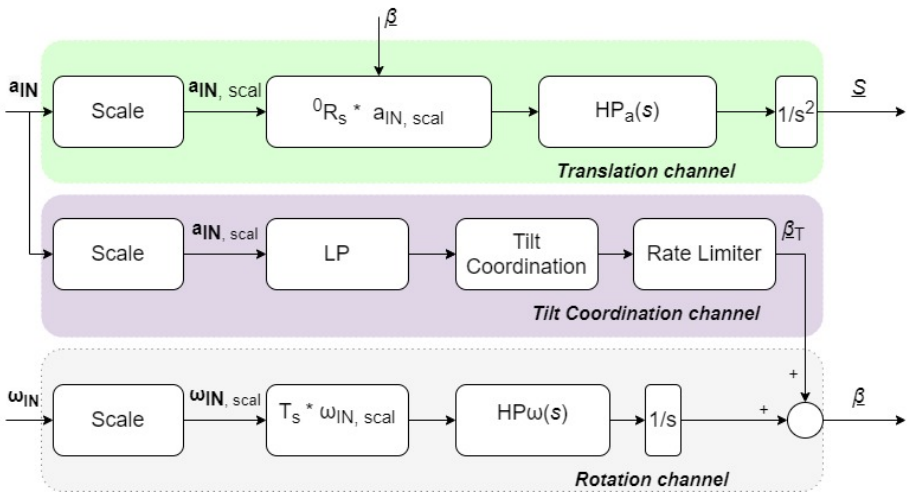

Fig. 6. Classical Washout filter (CWF) diagram [14].

CWF algorithm is simple to implement and computationally fast. However, it was designed for Stewart platform-based simulators. Therefore, it does not fit well for LaRRMoS, utilizing only a small fraction of the available workspace. To improve the effective workspace of the simulator, the CYL algorithm was implemented. The CYL

is a modified version of CWF that operates within a cylindrical coordinate system instead of a Cartesian one. The lateral motion of CWF is represented by rotational motion in CYL as shown in Fig. 7, greatly increasing the effective workspace of the simulator.

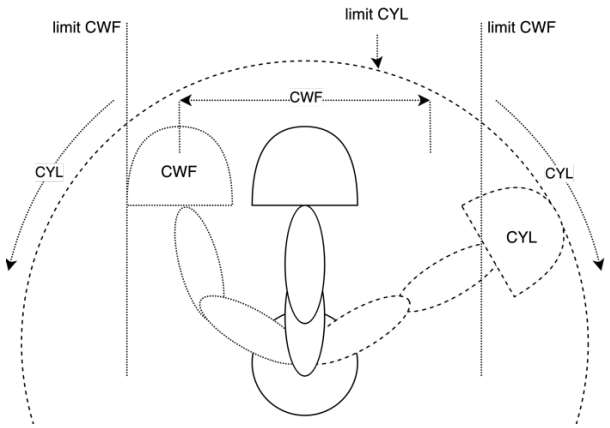

Fig. 7. Motion limits of the LaRRMoS using CWF and CYL algorithms [20].

The paper [14] describes the implementation of the CYL algorithm, suggested by [19], in LaRRMoS. Figure 8 shows a diagram of the CYL algorithm. To get cylindrical coordinates $\ddot{\xi}$ from Cartesian a_{IN}, a transformation matrix $C(\xi)$ is used with formula (5). Additional washout frame ${}^{W}R_S$ is introduced so that the axis of the gondola is always oriented with the axis of the washout filter. Besides operating in the cylindrical coordinate system, the new inertial compensation block is added. It is needed to compensate for centrifugal and Coriolis forces occurring during rotation around the A1 axis of the robot.

$$\ddot{\xi} = \begin{bmatrix} R \\ \alpha \\ Z \end{bmatrix} = C(\xi) \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{R} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (5)$$

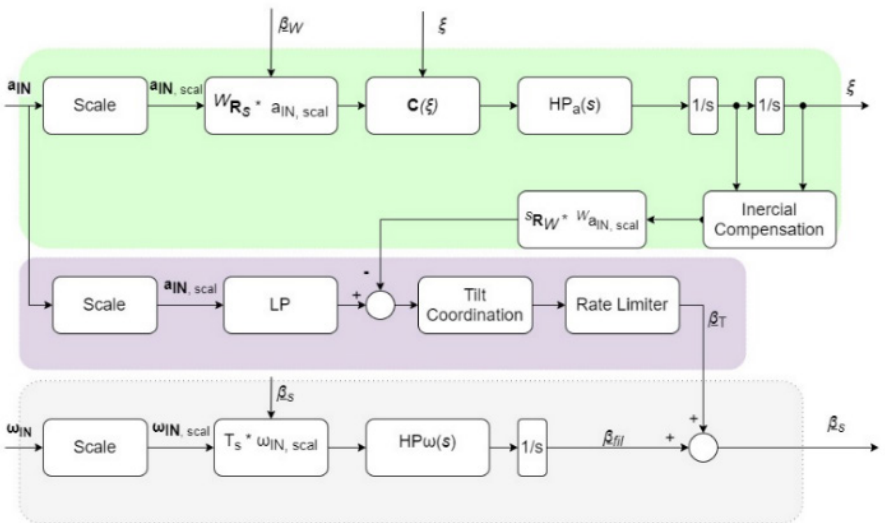

Fig. 8. Diagram of CYL algorithm [14].

There are MCAs that can achieve much better fidelity than CWF in simulating specific forces like model predictive control [21], [22], optimal [23], adaptive [24] and neural network-based [25]. However, most of them are computationally heavy, too slow or require complete knowledge of the future as stated in [26]. Though this study uses a pre-recorded set of motion data and runs simulations in non-interactive mode, it aims for real-time interactive simulation for LaRRMoS in the future, so only MCAs capable of real-time operation are inspected. Tuned parameters for CWF and CYL are presented in Table 1. The tuning process is described in detail in [14].

Table 1. Tuned Parameters for CWF and CYL Algorithms [20]

Parameter	CWF	CYL
Translation channel		
k_x	0.2	0.2
k_y	0.2	1.1
ζ_{HP}	1	1
ω_{nHP}	1	1
ω_b	1	1
Tilt Coordination		
k_{TCx}	0.5	0.5
k_{TCy}	1	1
ζ_{LP}	1.5	1.5
ω_{nLP}	8	8
Rotation channel		
k_{ROTx}	0.5	0.5
k_{ROTy}	1	1

5. TESTING METHODOLOGY

Both CWF and CYL algorithms were tested in two cases – car simulation and luge sports simulation. The car case is a typical choice for motion simulation as driving simulators are a common application. The luge sport case is chosen for its high dynamics, large roll, pitch, yaw angles, and rapidly changing accelerations. The assumption is that LaRRMoS is better suited for the simulation of highly dynamic vehicles, i.e., better in terms of motion cueing precision and effective workspace utilization. This assumption is based on research by [19], where Formula 1 simulation was developed for serial manipulator-based motion platform and also research by [27], where the same type of manipulator was used for upset recovery training of the pilots which involved manoeuvres that were not typical of classic flight simulators.

The method for examining the fidelity of the MCAs is based on a comparison of MCA input and output converted into specific forces as shown in Fig. 9. The sensation error is a measure of a user experienced difference between a real vehicle and a simulated one. The output of MCA is the robot end-effector's position and orientation. To compare it with the vehicle acceleration a_{in}, the second-order derivative of position S is calculated resulting in an acceleration a_{out}. However, to get specific force from acceleration a_{out} formula (6) is used as suggested by the [28], where θ and φ are roll and pitch angles of MCA output angles β.

$$SF = \begin{bmatrix} a_{Xout} + g \cdot sin(\theta) \\ a_{Yout} - g \cdot \sin(\varphi)\sin(\theta) \\ a_{Zout} + g \cdot \cos(\varphi)\sin(\theta) \end{bmatrix}. \quad (6)$$

For the luge simulation case, lateral acceleration a_{Yout} is represented by rotational motion, not linear. The formula changes to (7) for Y-axis acceleration, where angular acceleration $\Delta\omega/\Delta t$ of manipulator axis A1 is multiplied by radius R:

$$SF_Y = \frac{\Delta\omega}{\Delta t} \cdot R. \quad (7)$$

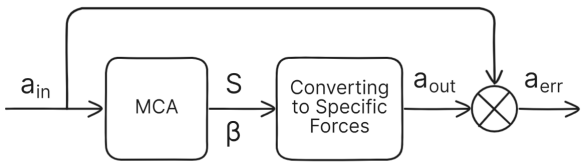

Fig. 9. MCA testing methodology.

6. RESULTS

6.1. Car Simulation Case

While theoretically, the calculated Y-axis acceleration is $2.48\frac{m}{s^2}$, recorded acceleration shows a gradual increase of the Y-axis acceleration up to $3.43\frac{m}{s^2}$, the acceleration increases because of the collapsing suspension (springs, tyres etc.). The difference between theoretical and measured (uncompensated) accelerations is $0.95\frac{m}{s^2}$. This substantial difference is caused by the gravity vector acting on a Y-axis acceleration due to the roll angle reaching up to 7.19°. Compensating Y-axis acceleration taking roll angle into account gives a Y-axis acceleration value of $2.22\frac{m}{s^2}$, thus reducing the error caused by gravity vector interference by 73 %. Additional error could be caused by imprecise measurement of the speed of the car entering the curve since the speed was measured and maintained by the speedometer of the car.

According to the proposed methodology, the output of the MCAs is compared with the car's specific force. The results are shown in Fig. 10 only for lateral acceleration. The output of CWF and CYL algorithms was acquired in two modes – with and without a tilt rate limiter. The error between the simulated acceleration and real vehicle acceleration is much more notable when the tilt rate limiter is used, compared to unrestricted tilt motion. Even without the tilt rate limiter, the additional error between a vehicle's recorded accelerations and simulated accelerations comes from frequency filter phase delay and tilt coordination angle limit of 30°. Also, an inverse kinematics solver theoretically could introduce additional error due to slight deviations of the robot path from the planned path to avoid singularities. However, in practice, the deviation measured with *Kuka.Trace* functionality was negligible [14].

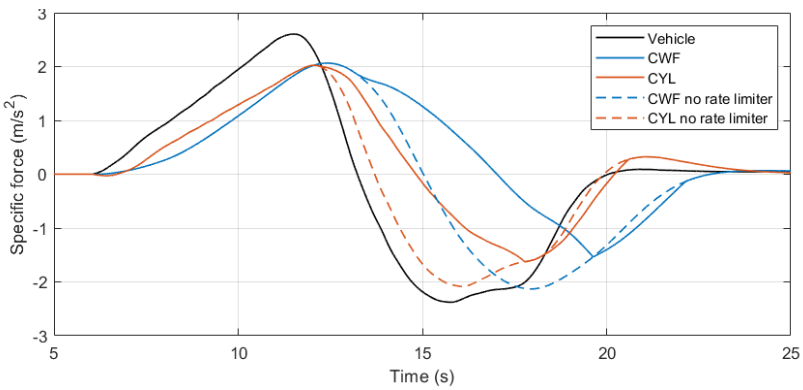

Fig. 10. MCA output compared with MCA input [14].

6.2. Luge Simulation Case

Extensive results for the luge sport case are published in a case study [20]. Recorded accelerations and orientations show high dynamics of the luge. Rapid changes in luge orientation result in both significant translational and angular accelerations. During the lap roll angle reaches $\pm 100\,°$, angular velocity reaches values up to $200\,°/s$, sustained linear accelerations reach values up to $\pm 2\,g$.

The highest accelerations appear on the Z-axis, and they are the most prolonged of all. Z-axis acceleration is a result of a luge sliding on a vertical surface of the track, so the curvature of the track generates acceleration up to $2\,g$ pushing the athlete onto the luge. The second significant acceleration arises on the Y-axis (sideways). While it changes rapidly, it is lower than the Z-axis acceleration and lies within the $\pm 1\,g$ range. The X-axis acceleration is the least aggressive and does not exceed $\pm 0.6\,g$. It appears due to the luge accelerating during straight sections of the track and decelerating on the curves.

For the luge simulation case, accelerations perceived in the simulator are compared with the desired accelerations of the luge in Fig. 11 and Fig. 12 for CWF and CYL, respectively. Target acceleration is the recorded acceleration of the luge used as an input for an MCA. Simulated acceleration is an acceleration perceived by the driver during the simulation. It consists of accelerations produced by the Translation channel (high-frequency component) and by the Tilt Coordination channel (low-frequency component).

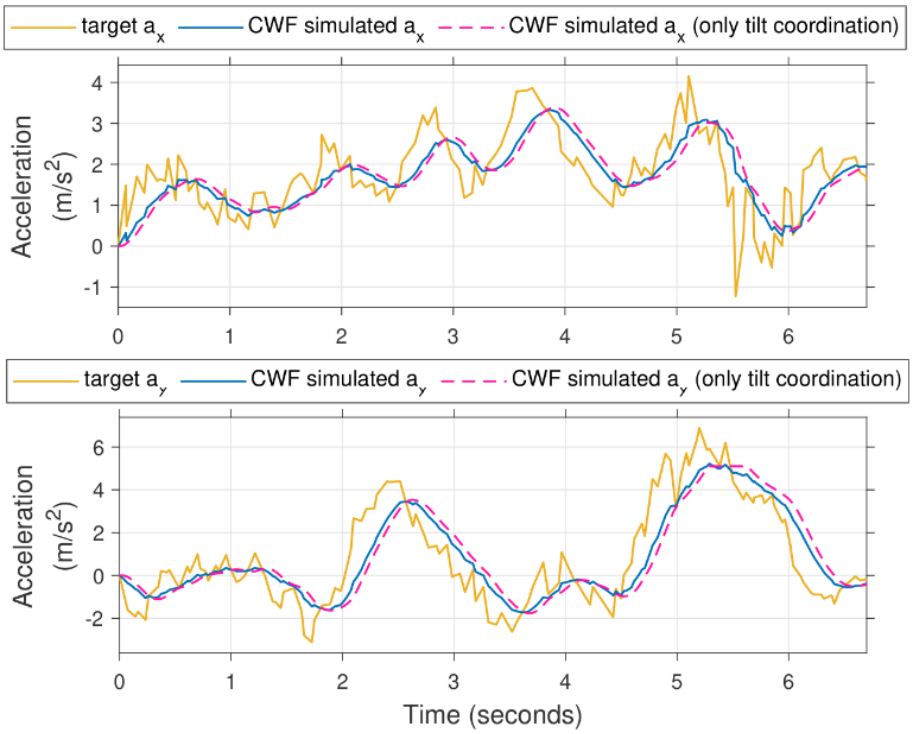

Fig. 11. CWF output compared to the input (target) [20].

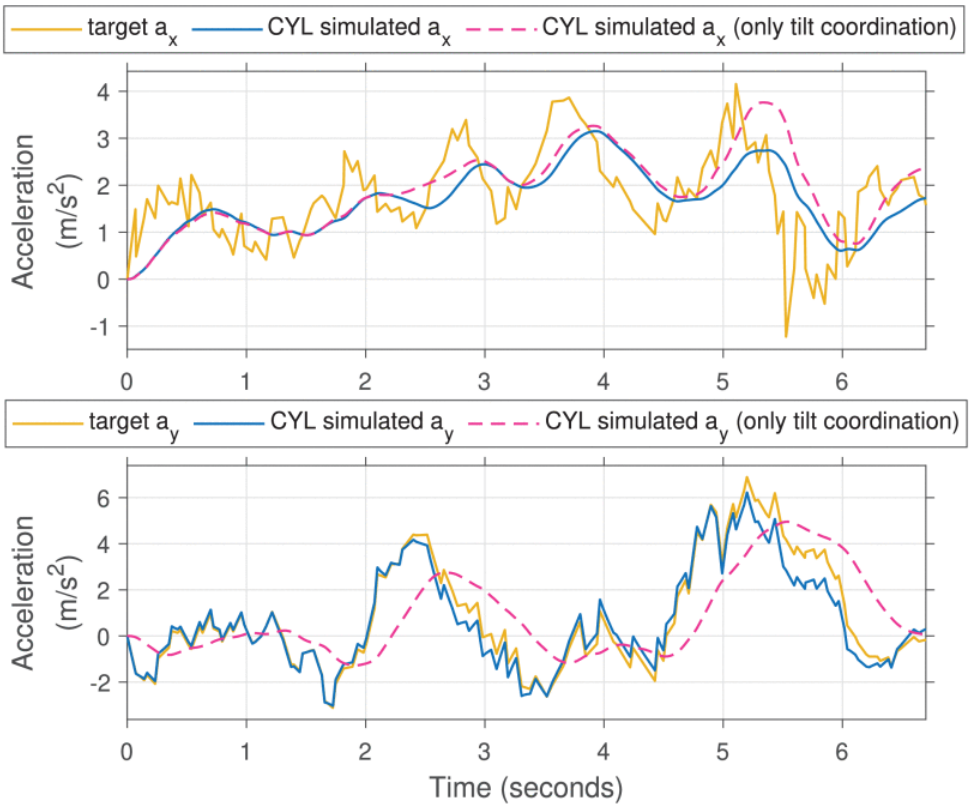

Fig. 12. CYL output compared to the input (target) [20].

In Figs. 11 and 12, the tilt coordination channel output is shown separately without translational channel output to indicate the significance of the high-frequency component. The continuous accelerations can be reproduced only using Tilt Coordination. However, the low-pass filter introduces the delay between the target and simulated acceleration. To compensate for the delay high-frequency acceleration is added. It is clearly visible in Fig. 12 where simulated acceleration matches target acceleration only when both high and low-frequency accelerations are combined.

Both MCAs produce imprecise X-axis acceleration. The reason is a double limitation – Tilt Coordination gain and the Translation channel gain are scaled down due to LaRRMoS physical limitations.

Y-axis acceleration is evidently better reproduced by the CYL algorithm, as it allows for wider and faster lateral movements. However, there are some discrepancies caused by the ±30 ° tilting limit and high-pass and low-pass filters.

7. DISCUSSION

While this paper addresses only motion cueing algorithms and vehicle dynamics acquisition, work on inverse kinematics solver feedback integration into the motion cueing algorithm, and implementation of real-time closed-loop motion control are

being conducted. Further research paper is planned to be published in the near future.

Only CWF and CYL algorithms were considered in the paper due to their computational simplicity and ability to work in real time. However, as the paper [26] states, some limited adaptive optimization can be added to the filter parameters to utilize the workspace of the robot even more efficiently. Introducing optimal filter MCA to the LaRRMoS control system will be investigated further.

Regarding the luge sport case discussed in this paper, obtaining the track geometry is essential to recreate it in a virtual environment for an interactive simulation. This will advance the training experience for athletes, improving their control of the luge during both simulation and real training. To achieve this goal, the GPS-RTK functionality could be added to the motion parameter acquisition system, as it greatly increases positioning precision. However, the lack of a stable GPS signal on a track has to be taken into account.

8. CONCLUSIONS

In this study, a motion control system for Large Range Robotic Motion Simulator has been proposed and tested. The findings prove that this system is effective in replicating provided vehicle-induced specific forces, especially for highly dynamic cases like luge sport. The motion cueing algorithms were tested in non-interactive mode with pre-recorded motion parameters using a body motion acquisition system, however, literature end experimental results show the computational simplicity of the algorithms indicating their ability to work in real-time interactive mode.

The Classical Washout Filter and its modification with the Cylindrical Coordinate System were tested for their motion cueing precision. The CYL algorithm provided better results in terms of utilizing the LaRRMoS workspace, especially for lateral accelerations. The luge simulation case highlighted the effectiveness of CYL compared to the CWF in highly dynamic motion replication. CYL algorithm is considered more effective for applications which tend to simulate more dynamic vehicles, like the luge sports examined in this study. Extrapolating on this assumption, CYL could be effectively used in racing car and motorcycle simulation or other extreme sport application. The CWF remains a viable option for simpler and less dynamic simulations.

Tilt rate limiters significantly increase motion cueing error. Removing the limiters significantly reduces the deviation between the target and simulated specific forces. The primary sources of the error are the delays produced by filtering, tilt coordination angle limits and physical limitations of LaRRMoS. The inverse kinematics solver, though, did not induce a noticeable error for the planned path of robots, thus indicating its high effectiveness in path execution.

REFERENCES

1. Nieuwenhuizen, F., Mulder, M., van Paassen, M., & Bülthoff, H. H. (2011). The Influence of motion system characteristics on pilot control behaviour. In *AIAA Modeling and Simulation Technologies Conference*, (pp. 1–15), 8–11 August 2011. Portland, Oregon, American Institute of Aeronautics and Astronautics.

2. Mohajer, N., Abdi, H., Nelson, K., & Nahavandi, S. (2015). Vehicle Motion Simulators, a Key Step towards Road Vehicle Dynamics Improvement. *Veh Syst Dyn, 53*, 1204–1226.
3. Caffò, A.O., Tinella, L., Lopez, A., Spano, G., Massaro, Y., Lisi, A., … & Bosco, A. (2020). The Drives for Driving Simulation: A Scientometric Analysis and a Selective Review of Reviews on Simulated Driving Research. *Front Psychol., 11*, 917.
4. Ghafarian, M., Watson, M., Mohajer, N., Nahavandi, D., Kebria, P.M., & Mohamed, S. M. (2023). A Review of Dynamic Vehicular Motion Simulators: Systems and Algorithms. *IEEE Access, 11*, 36331–36348.
5. Wynne, R.A., Beanland, V., & Salmon, P.M. (2019). Systematic Review of Driving Simulator Validation Studies, *Saf. Sci., 117*, 138–151.
6. Stewart, D. (1965). A Platform with Six Degrees of Freedom. *Proc. Inst. Mech. Eng., 180*, 371–386.
7. Mamoon, M., & Saifullah. (2014). Inverse kinematics and path planning of Stewart Platform using crank arm actuation system. In *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST),* (pp. 175–181), 14–18 January 2014. Islamabad, Pakistan.
8. Khamsehei Fadaei, M.H., Hamzeh Pahnehkolaei, S.A., Hesarlou, M.J., & Torkan, Z. (2017). Dynamics modeling of a stewart platform in Simulink MSC ADAMS. In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, (pp. 0294–0298).
9. Sharma, K., Haddadin, S., Minning, S., Heindl, J., Bellmann, T., Parusel, S., … & Albu-Schaeffer, A. (2013). Evaluation of human safety in the DLR Robotic Motion Simulator using a crash test dummy. In *2013 IEEE International Conference on Robotics and Automation*, 205–212.
10. Chance, G., Ghobrial, A., McAreavey, K., Pipe, T., & Lemaignan, S. (2022). On Determinism of Game Engines Used for Simulation-Based Autonomous Vehicle Verification. *IEEE Trans. Intell. Transp. Syst., 23*, 20538–20552.
11. Seefried, A., Pollok, A., Kuchar, R., Hellerer, M., Leitner, M., Schallert, C., … & Looye, G. (2019) Multi-Domain Flight Simulation with the DLR Robotic Motion Simulator. *2019 Spring Simulation Conference (SpringSim)*, 1–12.
12. Rassõlkin, A., Maksimkins, P., Stupāns, A., Rjabtšikov, V., Šenfelds, A., & Kuts, V. (2024). The Spatial Representation of a Self-Driving Vehicle for the Virtual Entity of a Digital Twin. *Computer, 57*, 58–66.
13. Maksimkins, P., Šenfelds, A., Maksimkina, J., & Ribickis, L. (2021). Development of a New Body Motion Parameters Acquisition System. In *2020 IEEE 8th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, (pp. 103–107), 22–24 April 2020. Vilnius, Lithuania.
14. Maksimkins, P., Stupāns, A., Kriviša, S., Šenfelds, A., & Ribickis, L. (2021). Implementation of a classic washout filter for robotic large range motion simulator using cylindrical coordinate system. *2021 IEEE 9th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, (pp. 1–6), 25–26 November 2021. Riga, Latvia.
15. Correia Grácio, B.J. (2013). *The Effects of Specific Force on Self-Motion Perception in a Simulation Environment.* PhD Thesis. TU Delft Repository.
16. Stahl, K., Leimbach, K.-D., Zoellner, R. D., & Meroth, A. (2015). A Washout and a Tilt Coordination Algorithm for a Hexapod Platform. *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 1196–1201.
17. Nahon, M.A., & Reid, L.D. (1990). Simulator Motion-Drive Algorithms - A Designer's Perspective. *J. Guid. Control Dyn., 13*, 356–362.
18. Reid, L., & Nahon, M. (1987). The Response of Airline Pilots to Flight Simulator Motion. *Flight Simulation Technologies Conference*. American Institute of Aeronautics and Astronautics.
19. Robuffo Giordano, P, Masone, C., Tesch, J., & Breidt, M. (2010). A Novel Framework for Closed-Loop Robotic Motion Simulation - Part II: Motion Cueing Design

and Experimental Validation. *2010 IEEE International Conference on Robotics and Automation*, 3896–3903.

20. Stupāns, A., Maksimkins, P., Šenfelds, A., & Ribickis, L. (2023). Analysis of motion cueing performance of large range robotic motion simulator: A luge sports simulator case study. In *2023 IEEE 17th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)*, (pp. 1–6), 14–16 June 2023. Tallinn, Estonia.
21. Chalak Qazani, M.R., Asadi, H., & Nahavandi, S. (2019). A Decoupled Linear Model Predictive Control-based Motion Cueing Algorithm for Simulation-based Motion Platform with Limited Workspace. *2019 IEEE International Conference on Industrial Technology (ICIT)*, 35–41.
22. Augusto, B., & Loureiro, R. (2009). *Motion Cueing in the Chalmers Driving Simulator: A Model Predictive Control Approach.* Chalmers University of Technology. Available at https://publications.lib.chalmers.se/records/fulltext/98871.pdf
23. Chen, S.-H., & Fu, L.-C. (2010). An optimal washout filter design for a motion platform with senseless and angular scaling maneuvers. In *Proceedings of the 2010 American Control Conference*, (pp. 4295–4300). Baltimore, MD, IEEE.
24. Miunske, T., Holzapfel, C., Baumgartner, E., & Reuss, H.-C. (2019). A new approach for an adaptive linear quadratic regulated motion cueing algorithm for an 8 DoF full motion driving simulator. In *2019 International Conference on Robotics and Automation (ICRA)*, (pp. 497–503), 20–24 May 2019. Montreal, Canada.
25. Koyuncu, A.B., Erçelik, E., Comulada-Simpson, E., Venrooij, J., Kaboli, M., & Knoll, A. (2020). A novel approach to neural network-based motion cueing algorithm for a driving simulator. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, (pp. 2118–2125), 20–23 October 2020. Las Vegas, USA.
26. Pham, D.A., Röttgermann, S., Flores, F.G., & Kecskemethy, A. (2015). Optimal motion cueing algorithm selection and parameter tuning for sickness-free robocoaster ride simulations. In B. Corves, E.-C. Lovasz, & M. Hüsing (Eds.), *Mechanisms, Transmissions and Applications* (pp. 127–135). Cham, Springer International Publishing.
27. Nie, Y., Bellmann, T., Seefried, A., Looye, G., Van Kampen, E.-J., & Chu, Q.P. (2015). Aircraft upset and recovery simulation with the DLR robot motion simulator. In *AIAA Modeling and Simulation Technologies Conference*, (pp. 1–10), 5–9 January 2015. Florida, American Institute of Aeronautics and Astronautics.
28. Kim, M.S., Moon, Y.G., Kim, G.D., & Lee, M.C. (2010). Partial Range Scaling Method Based Washout Algorithm for a Vehicle Driving Simulator and its Evaluation. *Int. J. Automot. Technol., 11*, 269–275.