

A Study on Vehicle Suspension Loads Prediction Method Based on Hybrid Road Simulation using Modelica Library and FMI

Minsu Hyun¹

¹Hyundai Motor Company, South Korea, mshyun@hyundai.com

Abstract

This study presents a method for predicting vehicle suspension component loads at the early design stage. A hybrid road simulation combines road load data acquired from a reference vehicle with the Time Waveform Replication (TWR) technique to generate virtual equivalent road profiles. The TWR was implemented in Python, and a multibody dynamics vehicle model developed using Modelon's Vehicle Dynamics Library was used to simulate chassis response. Integration and iterative simulation between the TWR system and the vehicle model were conducted via Functional Mock-up Units using the Python FMI library, FMPy. These virtual inputs were applied to a virtual test rig. In this study, road load data from a reference vehicle were used to derive the input signals, which were then applied to simulate the suspension loads of a target vehicle. Simulation results were validated against measurement data to confirm the effectiveness of the proposed method.

Keywords: Suspension Component Loads, Hybrid Road Simulation, TWR(Time Waveform Replication), Four-Post Test Rig, Virtual Testing Lab, Virtual Proving Ground, FMI(Functional Mock-up Interface), FMPy, Modelica Library

1 Introduction

In the early stages of vehicle development, where physical test vehicles are not available, it is essential to predict the loads input into suspension components to evaluate the durability performance of the suspension system. The predicted suspension component loads are used not only for evaluating the severity and life expectancy of each suspension part, but also for the development of durability test modes at the vehicle level, as well as test methods at the system, module, and component levels. For this reason, accurate prediction of suspension component loads is critical, and technologies for such predictions have been actively researched for decades.

As shown in Figure 1, the technologies used to predict suspension component loads can be broadly categorized into two types.

The first is the Virtual Testing Lab (VTL) approach, where measured wheel loads are directly applied to the wheel center of a fixed body model to predict component loads. This method has the advantages of simple analysis and the ability to use realistic drive signals, but it cannot consider changes in wheel loads due to design modifications or interactions between front and rear suspensions.

In contrast, the Virtual Proving Ground (VPG) method predicts component loads without any measured data by utilizing highly accurate road, tire, driver, and vehicle models. Although it can be applied to any type of vehicle, its accuracy heavily depends on the fidelity of the tire, road, and driver models, and it is difficult to accurately replicate the driving path, speed, and steering of a real test driver, making the method complex to operate.

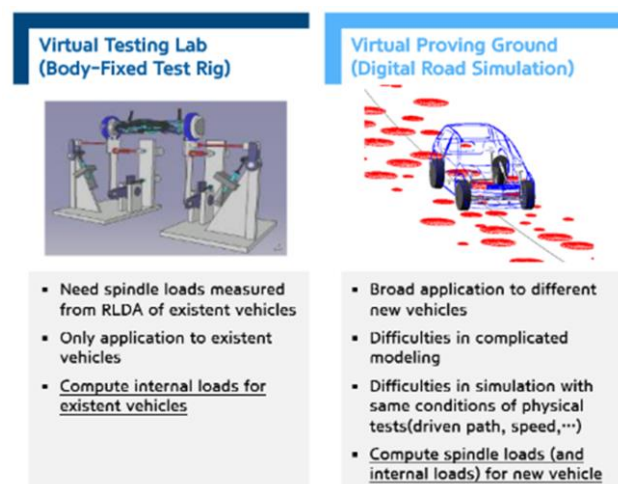


Figure 1. Types of component load prediction simulations.

This study proposes a hybrid road simulation-based suspension load prediction method that overcomes the limitations and leverages the strengths of both VTL and VPG. Section 2.1 introduces the fundamental concept of hybrid road simulation using FMI and Modelica Library. Section 2.2 explains the Time Waveform Replication (TWR) algorithm required for hybrid road simulation. Section 2.3 presents the virtual test rig and simulation method. Sections 2.4 and 2.5 present case studies using

RLDA (Road Load Data Acquisition) data from the reference vehicle and the target vehicle to validate the proposed methodology.

2 Methodology

2.1 Overview of Hybrid Road Simulation

Hybrid road simulation is a technique that utilizes both RLDA-measured data and a vehicle model. As shown in Figure 2, the core process of hybrid road simulation involves generating equivalent virtual road profiles or drive signal on a virtual test rig to replicate target signals from RLDA data using the simulation model of the reference vehicle. The inputs generated are then applied to the simulation model of the target vehicle to predict component loads. This approach should be applied to similar platforms to ensure accuracy, but it offers the advantages of simulating realistic responses without requiring separate road or driver models, thus reducing model dependency.

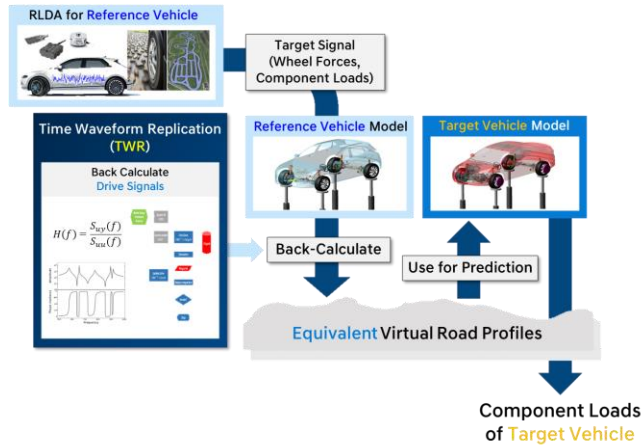


Figure 2. Hybrid road simulation process.

To perform hybrid road simulation, a TWR algorithm is required. The TWR methods include an iteration-based technique using Frequency Response Functions (FRF) and a control-based approach using PID or Iterative Learning Control (ILC). The FRF-based method requires more time due to its iterative nature but is easier to use and offers good reproducibility. The control-based method eliminates the need for iterations but requires optimization of the controller to improve tracking performance.

2.2 Time Waveform Replication (TWR)

In this study, the FRF-based iterative method among TWR techniques was employed to extract drive signals for the hybrid road simulation. Figure 3 illustrates the TWR process, which consists of four main steps: (1) FRF generation using white-pink noise, (2) initial drive signal creation, (3) iterative drive signal update, and (4)

convergence check using RMSE thresholds. Each step is described in detail in the following subsections.

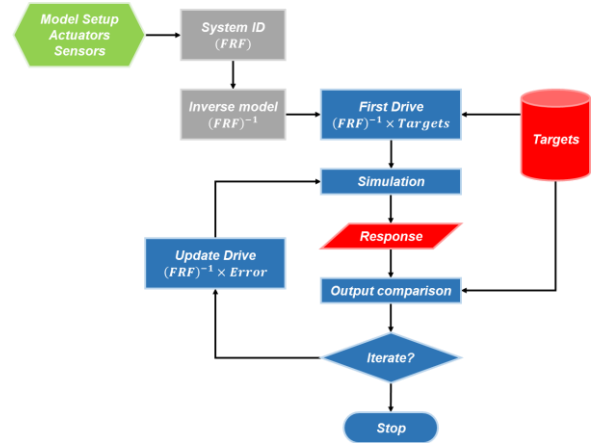


Figure 3. Time waveform replication (TWR) process.

2.2.1 FRF Generation Using White-Pink Noise

In the first step, white-pink noise is applied to the input channels of the vehicle model to extract responses from the output channels and generate an FRF model. The white-pink noise $WPN(f)$ has a constant magnitude from the start frequency (f_{st}) to the border frequency (f_b) and gradually decreases from f_b to the end frequency (f_{en}), as shown in Figure 4. Equation (1) defines $WPN(f)$ across these segments, where the attenuation after f_b depends on the exponential parameter p .

$$WPN(f) = \begin{cases} \left(\frac{1}{f_b}\right)^{2p}, & f_{st} \leq f < f_b \\ \left(\frac{1}{f}\right)^{2p}, & f \geq f_b \end{cases} \quad (1)$$

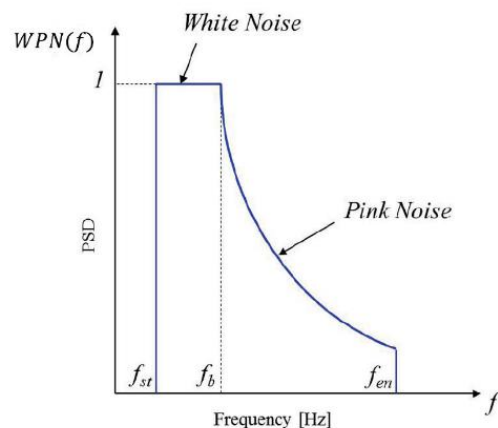


Figure 4. White-Pink Noise.

The process for FRF generation is shown in Figure 5.

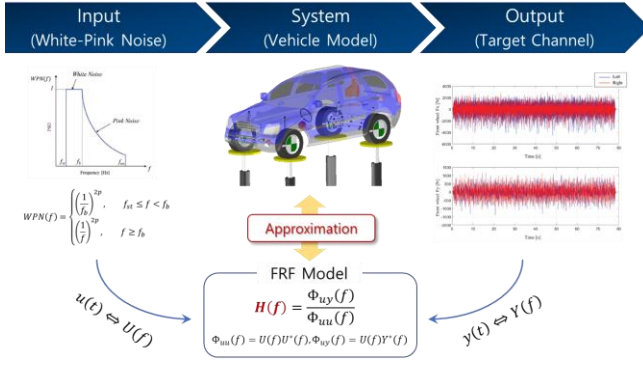


Figure 5. System identification (FRF generation) process.

In the FRF generation process, the auto-spectrum represents the frequency-domain power distribution of a signal, typically used to analyze the input characteristics. The cross-spectrum, on the other hand, quantifies the frequency-domain relationship between the input and output signals. These spectral components are used to compute the Frequency Response Function (FRF) by dividing the cross-spectrum by the auto-spectrum, enabling the identification of system dynamics. The cross-spectrum and auto-spectrum are obtained as shown in Equations (2) and (3):

$$\Phi_{uy}(f) = U(f)Y^*(f) \quad (2)$$

$$\Phi_{uu}(f) = U(f)U^*(f) \quad (3)$$

Here, $U(f)$ and $Y^*(f)$ are the Fourier transforms of $u(t)$ and $y(t)$, respectively, and $*$ denotes the complex conjugate. The FRF $H(f)$ is then derived as:

$$H(f) = \frac{\Phi_{uy}(f)}{\Phi_{uu}(f)} \quad (4)$$

2.2.2 Initial Drive Signal Generation

The initial drive signal is generated using the inverse of the FRF and the inverse Fourier transform as shown in Equation (5):

$$u(t) = d \cdot \mathcal{F}^{-1} \left(H(f)^{-1} \mathcal{F}(S \cdot w(t)) \right) \quad (5)$$

Here, $H(f)^{-1}$ is the inverse FRF, $\mathcal{F}$ is the Fourier transform, $w(t)$ is the target signal, S and d are weights for the target and drive signals, respectively. The signal is valid within linear system regions, but nonlinear behavior introduces error, necessitating additional correction.

2.2.3 Drive Signal Update via Iteration

To correct errors between target and actual responses, an updated drive signal is computed iteratively. Equation (6) represents the iterative update of the drive signal in the

TWR process. The initial drive signal is computed by applying the inverse FRF to the target signal. In subsequent iterations, the error between the simulated response and the target signal is calculated and passed through the inverse FRF to generate a correction signal. This correction is then added to the previous drive signal, progressively improving the match between the simulated and target responses. This approach is grounded in frequency-domain system identification and iterative learning control principles and has been widely used in road load replication studies for its convergence reliability and simplicity. The $k + 1$ -th drive signal is calculated as in Equation (6):

$$u_{k+1}(t) = u_k(t) + d \cdot \mathcal{F}^{-1} \left(H(f)^{-1} \mathcal{F} \left(E \cdot (y(t) - S \cdot w(t)) \right) \right) \quad (6)$$

Here, E is the error weighting parameter, and $y(t)$ is the output of the k -th iteration. The process continues until the RMSE (Root Mean Square Error) of each output channel is below the threshold.

2.3 Virtual Test Rig for Hybrid Road Simulation

The virtual test rig used in this study was constructed using a full vehicle model based on Modelon's Vehicle Dynamics Library (VDL), which supports high-fidelity multibody dynamics (MBD) simulation. This MBD-based model captures realistic vehicle responses and suspension loads by simulating the actual kinematics and compliance characteristics of suspension and chassis systems. The suspension architecture of the reference and target vehicles consists of a McPherson strut configuration at the front and a multi-link suspension at the rear, as illustrated in Figures 6 and 7. This combination was selected to reflect a typical modern passenger vehicle setup, where the front provides packaging efficiency and steering simplicity, while the rear enables better ride comfort and lateral stability.



Figure 6. Front suspension model (McPherson strut) in VDL.

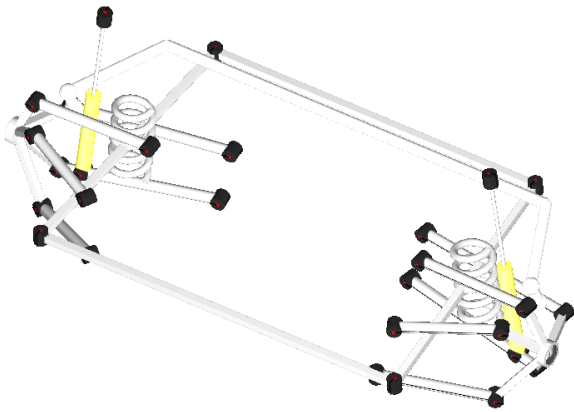


Figure 7. Rear suspension model (multi-link) in VDL.

While developing the suspension model, it was identified that the default bump stop element in Modelon's VDL lacks support for nonlinear force-displacement characteristics, which are critical for accurately capturing suspension component loads during durability simulations. To address this limitation, a custom bump stop model was implemented using Modelica. This component incorporates a user-defined nonlinear force-displacement curve using a lookup table and becomes active when the damper compression exceeds a predefined bump clearance. This modeling approach enables accurate replication of the sharp load increase that occurs during bottoming events, which significantly influences suspension component fatigue and durability assessment. To support this, two dedicated submodels were constructed and are illustrated in Figures 8 and 9.

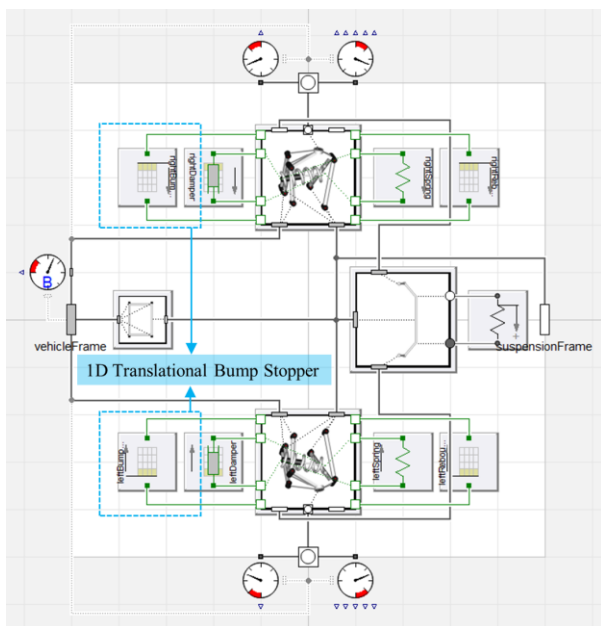


Figure 8. 1D translational bump stopper model.

Figure 8 shows the custom 1D translational bump stopper model used in the rear suspension system. The element accepts the force-displacement curve and is activated only when bump clearance is exceeded.

Figure 9 presents the kinematic model in the rear suspension linkage model that computes the bump stop compression displacement. In this configuration, the bump stop is mounted not on the shock absorber upper bushing but directly on the vehicle body. As such, the damper stroke alone does not reflect bump stop engagement. Therefore, an additional kinematic model was developed to calculate the relative displacement between the lower control arm and the chassis hardpoint. This value serves as the effective compression input to the bump stop force model.

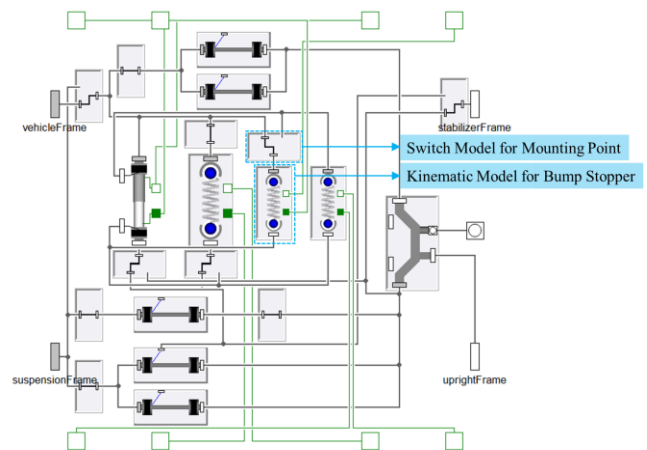


Figure 9. Kinematic model for bump stopper.

This approach ensures that the bump stop responds only under physically accurate conditions, enhancing the realism of load prediction and its influence on suspension durability performance.

The virtual test rig is configured as a 4-post setup, capable of applying vertical displacements at the tire contact patches through equivalent virtual road profiles. Additionally, actuators are installed at each wheel center to apply longitudinal and lateral forces, as well as moments. This configuration enables comprehensive reproduction of real-world wheel center load conditions and supports the simulation of both road-induced and control-induced excitations. Figure 10 shows the layout of this virtual durability test rig within the simulation environment.

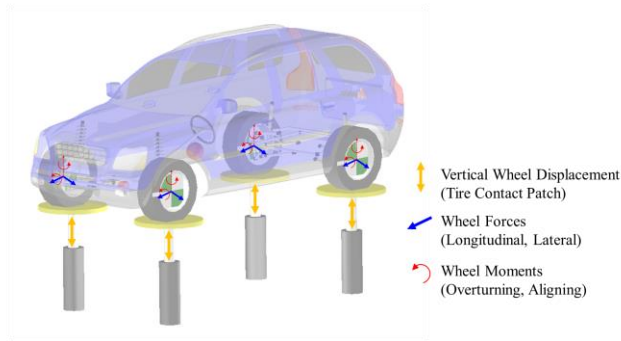


Figure 10. Configuration of virtual test rig.

To enable hybrid road simulation, the entire vehicle model and test rig were exported as Functional Mock-up Units (FMUs) using the FMI standard. These FMUs were integrated with a Python-based TWR algorithm, implemented using the FMPy library. This setup allows for automated iteration of drive signals to match target RLDA responses. The overall simulation framework and data flow between Python and the FMU-based vehicle model are depicted in Figure 11.

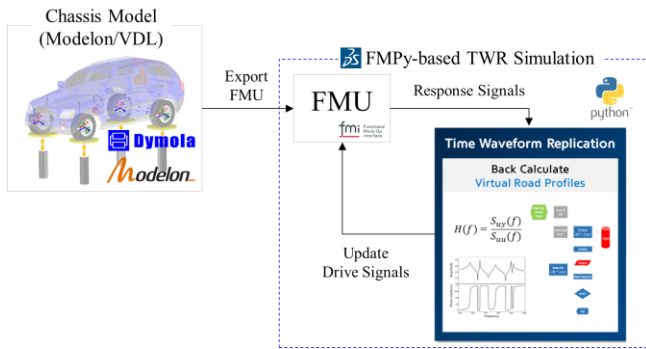


Figure 11. TWR simulation based on FMPy.

The TWR algorithm operates by comparing the actual response of the virtual vehicle to the target RLDA signals. It continuously updates the drive signals by feeding new input signals, stored in *.mat format, into the FMU simulation. These updates are computed using an FRF-based iterative optimization approach, which enhances tracking fidelity for complex time-domain signals. During each TWR iteration, the Python controller:

Step 1. Executes a simulation with the current drive signal.

Step 2. Evaluates the error between target and simulated output (e.g., wheel center forces).

Step 3. Updates the drive signal using the inverse FRF and re-simulates until convergence.

This tightly integrated Python-FMI workflow provides robust co-simulation performance while ensuring that realistic

suspension dynamics are captured through the MBD-based virtual rig.

2.4 Application to Hybrid Road Simulation

2.4.1 Drive Signal Update via Iteration

For the reference vehicle, equivalent virtual road profiles and drive signals capable of reproducing 20 channels of wheel forces and moments were generated based on Belgian road RLDA data. Predicted and measured component and wheel loads were compared. Figures 12 and 13 show the improvement in front and rear vertical wheel force prediction accuracy across iterations. Figure 14 illustrates the RMSE convergence behavior across 10 iterations for 20 wheel load channels. All channels showed a consistent downward trend in RMSE, indicating stable convergence. While convergence patterns were similar overall, the final RMSE values for vertical loads remained slightly higher, stabilizing below 12%. This level of accuracy is considered reasonable for durability simulation purposes. In contrast, wheel center longitudinal and lateral loads, as well as aligning and overturning moments, achieved lower residual RMSE values, typically below 6%. These results confirm the robustness of the FRF-based iterative TWR algorithm in minimizing signal error across diverse load types.

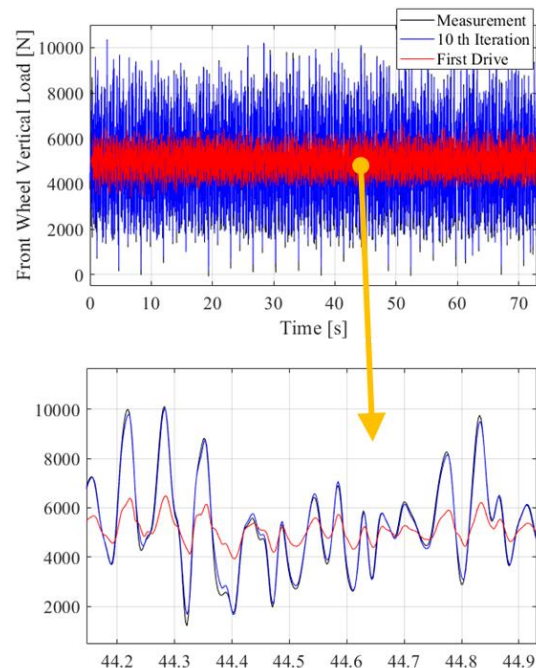


Figure 12. Predicted results of front wheel center vertical load.

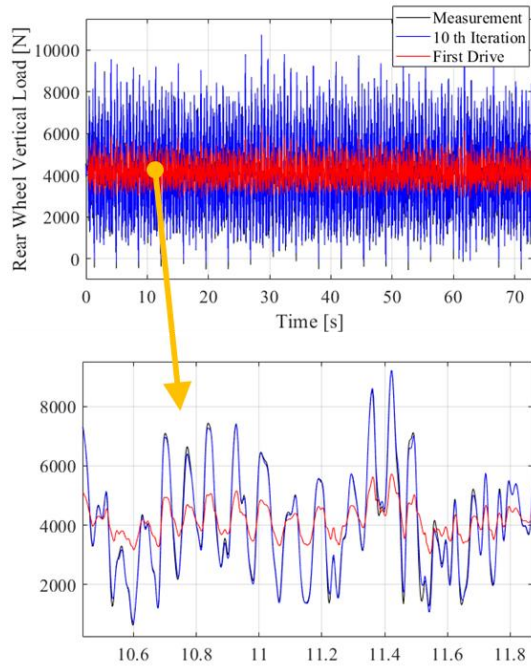


Figure 13. Predicted results of rear wheel center vertical load.

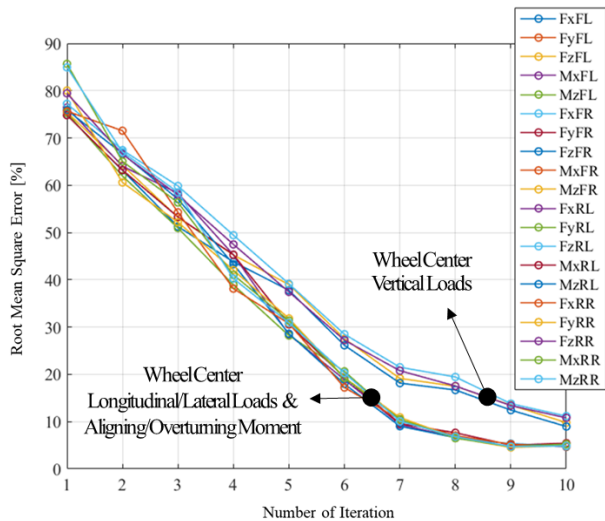


Figure 14. RMSE history of wheel loads and moments.

In this study, component damage was calculated as a normalized pseudo damage metric using rainflow cycle counting and Miner's rule. The damage values represent relative fatigue severity based on the simulated load time histories, allowing comparison between measured and predicted results without requiring material-specific S-N curves. Table 1 summarizes RMS and normalized pseudo damage values of measured and simulated component loads, confirming the high accuracy of the generated drive signal and road profile.

Table 1. TWR simulation results of reference vehicle.

Channel	Index	Simulation	Target	Error
Front Wheel Center Vertical Load	RMS	1729.62	1691.39	2.26 %
	Pseudo Damage	1.34	1.24	8.06 %
Rear Wheel Center Vertical Load	RMS	2074.84	2048.02	1.31 %
	Pseudo Damage	1.81	1.7	6.47 %
Front Lower Arm Boll Joint Fx Load	RMS	875.29	866.18	1.05 %
	Pseudo Damage	0.33	0.32	3.13 %
Real Assist Arm Axial Load	RMS	1634.37	1615.21	1.19 %
	Pseudo Damage	2.88	2.80	2.86 %
Rear Lower Arm Axial Load	RMS	1548.75	1571.01	1.42 %
	Pseudo Damage	2.80	2.96	5.40 %

2.4.2 Simulation for Target Vehicle

The extracted drive signals and road profiles were applied to the target vehicle model to predict component and wheel loads. Figures 15 to 19 present range pair graphs showing load distribution and cumulative damage. While vertical loads and front lower arm ball joint loads closely matched measured data, some discrepancies were observed in rear component loads, primarily due to sensitivity of damage metrics and reliance on reference vehicle measurements for lateral/longitudinal forces.

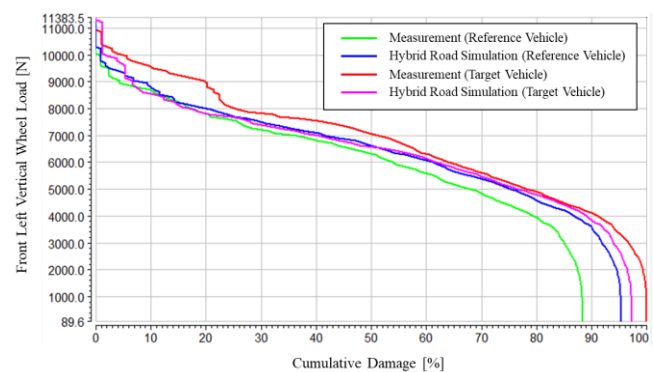


Figure 15. Damage of front wheel center vertical load.

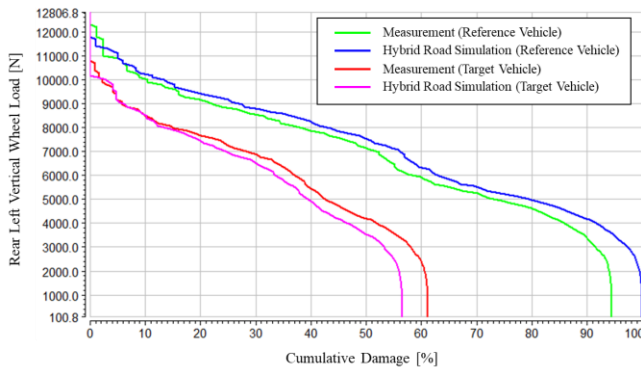


Figure 16. Damage of rear wheel center vertical load.

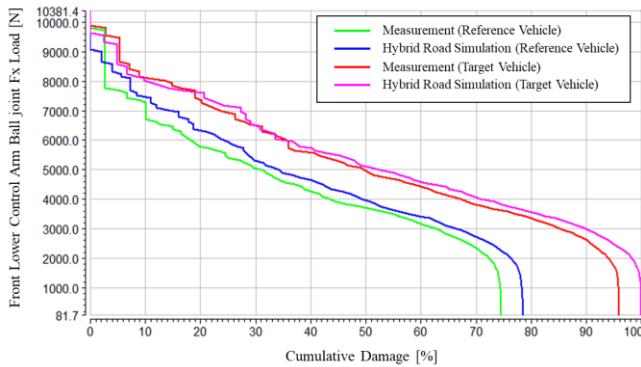


Figure 17. Damage of front lower arm ball joint load.

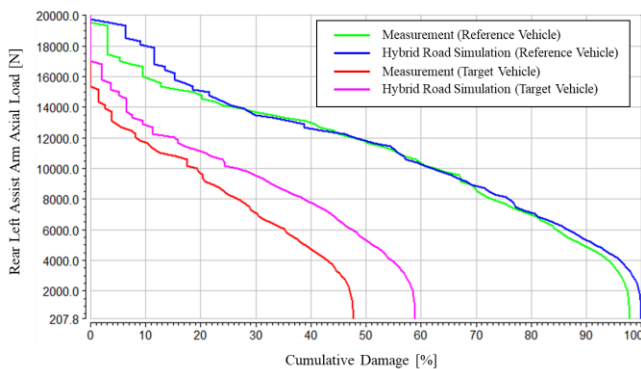


Figure 18. Damage of rear assist arm axial load.

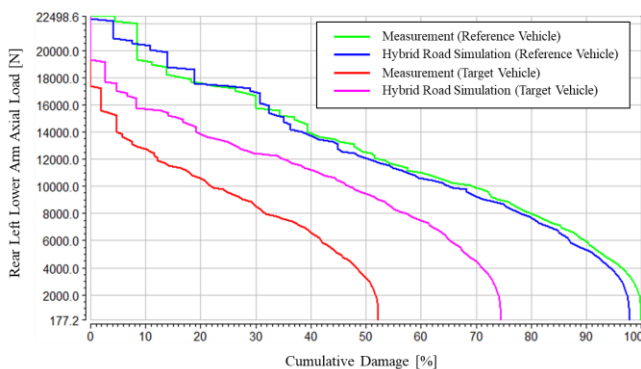


Figure 19. Damage of rear lower arm axial load.

Table 2 presents the RMS and damage values for major suspension components, showing that trends in damage and RMS variation were reproduced reasonably well even for the rear suspension.

Table 2. TWR simulation results of target vehicle.

Channel	Index	Simulation	Target	Error
Front Wheel	RMS	1726.78	1842.046	6.26 %
Center Vertical Load	Pseudo Damage	1.36	1.4	2.86 %
Rear Wheel	RMS	1785.68	1819.82	1.87 %
Center Vertical Load	Pseudo Damage	1.02	1.1	7.27 %
Front Lower Arm Boll Joint Fx Load	RMS	875.63	858.28	2.02 %
	Pseudo Damage	0.424	0.372	13.98 %
Real Assist Arm Axial Load	RMS	1291.52	1251.66	3.18 %
	Pseudo Damage	1.69	1.37	23.36 %
Rear Lower Arm Axial Load	RMS	1419.09	1345.80	5.45 %
	Pseudo Damage	2.40	1.75	37.14 %

3 Conclusion

This study presented a hybrid road simulation method for predicting vehicle suspension component loads by integrating road load data and multibody vehicle models through FMI-based co-simulation. By employing an FRF-based iterative TWR algorithm, equivalent virtual road profiles were generated using RLDA data from a reference vehicle and applied to a target vehicle model exported as an FMU.

The proposed methodology offers several advantages relevant to the Modelica and FMI community:

- **FMI-based modular integration:** The approach demonstrates a practical application of the FMI standard using FMPy to couple a Python-based TWR algorithm with a Modelica-based full vehicle model, ensuring seamless interoperability in a simulation workflow.
- **Efficient virtual durability testing:** By leveraging existing RLDA data and applying it through a virtual test rig configuration, the method enables early-stage durability analysis without the need for physical prototypes or detailed driver models.

• **System-level fidelity with manageable complexity:**

Compared to traditional Virtual Proving Ground (VPG) approaches, the hybrid method maintains accuracy in load prediction while reducing the dependency on high-fidelity subsystems like road or tire models.

Future work will focus on extending this FMI-based hybrid simulation framework to cover broader driving scenarios and enhancing prediction robustness, especially for design-dependent variables such as lateral and longitudinal load components. This research illustrates how the use of standardized model exchange formats like FMI can accelerate the development and validation of vehicle durability simulation methods in industrial applications.

References

- Schultz, R., & Carter, S. (2023). "A MIMO time waveform replication control implementation." Society for Experimental Mechanics Annual Conference and Exposition. Cham: Springer Nature Switzerland.
- Bäcker, M., Langthaler, T., Olbrich, M., Oppermann, H. (2005). "The Hybrid Road Approach for Durability Loads Prediction". SAE Technical Paper. No. 2005-01-0628
- Modelica Association (2013-07). *Modelica – A Unified ObjectOriented Language for Systems Modeling. Language Specification Version 3.2 Revision 2*. Tech. rep. Linköping: Modelica Association. URL: <https://www.modelica.org/documents/ModelicaSpec32Revision2.pdf>.
- Kugu, O., Zhou, S., Nowak, R., Müller, G., Reiterer, S. H., Meierhofer, A., ... & Grafinger, M. (2023). An fmi-and ssp-based model integration methodology for a digital twin platform of a holistic railway infrastructure system. In Modelica Conferences (pp. 717-726).
- Hatledal, L. I., Zhang, H., Styve, A., & Hovland, G. (2018). Fmi4j: A software package for working with functional mock-up units on the java virtual machine. In *The 59th Conference on Simulation and Modelling (SIMS 59)*. Linköping University Electronic Press, Linköpings universitet.
- Hyun, M., & Heo, S.-J. (2022). "Modular Architecture Design of Chassis System Using Database." *International Journal of Automotive Technology*, 23, 225–232. Cham: Springer Nature Switzerland. URL: <https://link.springer.com/article/10.1007/s12239-022-0019-9>
- Park, S., Jeon, Y., Kang, D.-o., Hyun, M., & Heo, S.-J. (2019). "Predicting the Vehicle Performance at an Early Stage of Development Process via Suspension Bushing Design Tool." *Modelica Conference 2019*. Linköping University Electronic Press. URL: https://ep.liu.se/en/conference-article.aspx?series=157&issue=157&Article_No=86
- Ko, K.-c., Park, S., Kim, J., Jeong, Y., Hyun, M., & Heo, S.-J. (2017). "Development of Hierarchal Commercial Vehicle Model for Target Cascading Suspension Design Process." *Modelica Conference 2017*. Linköping University Electronic Press. URL: https://ep.liu.se/en/conference-article.aspx?issue=132&Article_No=49