National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Optimization of a Full-Vehicle Suspension System using Unsteady Continuous Adjoint

Diploma Thesis

by

Dimitrios Mavromatis

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2026

Abstract

Designing passive vehicle suspension systems requires navigating a complex trade-off between conflicting objectives, such as ride comfort and road holding. While stochastic optimization methods are commonly used, they are often computationally expensive and slow to converge. Focusing strictly on ride comfort, this diploma thesis proposes a highly efficient, gradient-based optimization framework for an 8-degree-of-freedom (DoF) full-vehicle model, utilizing the Unsteady Continuous Adjoint Method to compute exact Sensitivity Derivatives. The core contribution of this work is the development of a C++ framework that integrates the primal equations of motion with their adjoint counterparts to compute the exact gradients of the objective function. These gradients are then utilized by a BFGS optimization algorithm to iteratively guide the design variables toward an optimal configuration. The sole objective of the optimization is to minimize driver discomfort. To avoid adding more equations to the system-which is required by standard ISO ride comfort metrics-a composite objective function was developed. This formulation utilizes the vertical jerk of the Driver's Seat to implicitly penalize high-frequency vibrations, combined with vertical, pitch, and roll accelerations. This approach maintains the original number of primal equations while effectively capturing subjective harshness. A significant difficulty in unsteady adjoint optimization is the high memory requirement for storing the primal state history. This limitation was overcome by implementing the Check-pointing algorithm, effectively reducing memory usage with only a marginal increase in computational cost. The framework was applied through three distinct driving scenarios: deterministic bumps, harmonic excitations, and stochastic road profiles. Numerical results demonstrate the algorithm's ability to rapidly converge to optimal configurations, achieving reductions of the objective function ranging from 62% to 89%. The findings confirm that the proposed framework offers a robust and memory-efficient solution for vehicle suspension optimization.

Keywords: Vehicle Dynamics, Suspension Optimization, Unsteady Continuous Adjoint Method, Check-pointing, Ride Comfort, C++, Gradient-based Optimization.

Περίληψη

Ο σχεδιασμός παθητικών συστημάτων ανάρτησης οχημάτων απαιτεί την εύρεση μιας ισορροπίας ανάμεσα σε αντικρουόμενους στόχους, όπως η άνεση των επιβατών και η οδική συμπεριφορά (κράτημα). Ενώ οι στοχαστικές μέθοδοι βελτιστοποίησης χρησιμοποιούνται ευρέως, είναι συχνά υπολογιστικά δαπανηρές και παρουσιάζουν αργή σύγκλιση. Εστιάζοντας αποκλειστικά στη οδηγική άνεση, η παρούσα διπλωματική εργασία προτείνει μια μέθοδο αιτιοκρατικής βελτιστοποίησης για ένα μοντέλο πλήρους οχήματος 8 βαθμών ελευθερίας, χρησιμοποιώντας τη Μη-Μόνιμη Συνεχή Συζυγή Μέθοδο για τον υπολογισμό των παραγώγων ευαισθησίας. Ο κώδικας σε γλώσσα C++ που αναπτύχθηκε ενσωματώνει τις αρχικές εξισώσεις κίνησης με τις αντίστοιχες συζυγείς τους για τον υπολογισμό των ακριβών κλίσεων της συνάρτησης κόστους. Στη συνέχεια, οι κλίσεις αυτές αξιοποιούνται από τον αλγόριθμο βελτιστοποίησης BFGS για την επαναληπτική καθοδήγηση των μεταβλητών σχεδιασμού προς μια βέλτιστη διαμόρφωση. Ο μοναδικός στόχος της βελτιστοποίησης είναι η ελαχιστοποίηση της δυσφορίας του αναβάτη. Για τον σκοπό αυτό αναπτύχθηκε μια σύνθετη συνάρτηση κόστους. Η εργασία αυτή χρησιμοποιεί την κατακόρυφη μεταβολή της επιτάχυνσης (jerk) του καθίσματος του οδηγού για τη μείωση των κραδασμών υψηλής συχνότητας. Επιπλέον, χρησιμοποιούνται τα μεγέθη επιτάχυνσης του καθίσματος του οδηγού, καθώς και των κινήσεων βύθισης/πρόνευσης και πλευρικής κλίσης του πλαισίου. Αυτή η προσέγγιση δεν προσθέτει επιπλέον εξισώσεις, ενώ αποτυπώνει αποτελεσματικά την υποκειμενική αίσθηση δυσφορίας. Μια σημαντική δυσκολία στη μη-μόνιμη συζυγή μέθοδο είναι οι μεγάλες απαιτήσεις μνήμης για την αποθήκευση ολόκληρης της ιστορίας των λύσεων. Ο περιορισμός αυτός ξεπεράστηκε με την εφαρμογή του αλγορίθμου Check-pointing, επιτυγχάνοντας μια δραματική μείωση της απαιτούμενης μνήμης με μόνο μια οριακή αύξηση του υπολογιστικού κόστους. Ο κώδικας χρησιμοποιήθηκε σε τρία διαφορετικά σενάρια οδοστρώματος: ένα μειωτή ταχύτητας (σαμαράκι), έναν δρόμο με αρμονικό/ημιτονοειδές προφίλ και ένα τυχαίο προφίλ οδοστρώματος. Τα αριθμητικά αποτελέσματα αναδεικνύουν την ικανότητα του επιλύτη να συγκλίνει ταχύτατα σε βέλτιστες διαμορφώσεις οχήματος, επιτυγχάνοντας μειώσεις της συνάρτησης κόστους που κυμαίνονται από 62% έως 89%, επιβεβαιώνοντας την αποτελεσματικότητα του.

Λέξεις-κλειδιά: Δυναμική Οχημάτων, Βελτιστοποίηση Ανάρτησης, Μη-Μόνιμη Συνεχής Συζυγής Μέθοδος, Check-pointing, Άνεση Επιβατών, C++, Αιτιοκρατικές Μέθοδοι Βελτιστοποίησης.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Kyriakos C. Giannakoglou, for the opportunity to work on this challenging subject and for his guidance throughout the course of this diploma thesis. His scientific insight was essential in navigating the complexities of the adjoint method. I am also grateful to Dr. Varvara Asouti for her willingness to help and her valuable advice regarding the technical implementation of the code.

Finally, I thank my family: my parents Tasos and Katerina, my brothers George and Elias, my grandparents, and my girlfriend Stella, for their support and endless patience during the long hours this work required.

This thesis is devoted to them.

Thank you.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον επιβλέποντα καθηγητή μου, κ. Κυριάκο Χ. Γιαννάκογλου, για την ευκαιρία που μου έδωσε να εργαστώ πάνω σε αυτό το απαιτητικό αντικείμενο, καθώς και για την καθοδήγησή του καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας. Η επιστημονική του οξυδέρκεια υπήρξε καθοριστική για την κατανόηση των πολυπλοκοτήτων της συζυγούς μεθόδου. Είμαι επίσης ευγνώμων στη Δρ. Βαρβάρα Ασούτη για την προθυμία της να βοηθήσει και για τις πολύτιμες συμβουλές της σχετικά με την τεχνική υλοποίηση του υπολογιστικού κώδικα.

Τέλος, ευχαριστώ την οικογένειά μου: τους γονείς μου Τάσο και Κατερίνα, τα αδέρφια μου Γιώργο και Ηλία, τους παππούδες μου, καθώς και τη σύντροφό μου Στέλλα, για τη στήριξη και την αμέριστη υπομονή που έδειξαν κατά τις ατελείωτες ώρες που απαιτήθηκαν για την ολοκλήρωση αυτής της εργασίας.

Η εργασία αυτή αφιερώνεται σε αυτούς.

Σας ευχαριστώ.

Contents

Chapter 1

Introduction

1.1 Vehicle Dynamics and Suspension

The suspension system serves as the mechanical link connecting the vehicle body (sprung mass) to the wheels (unsprung masses). Its design dictates how the vehicle interacts with the road, fulfilling two distinct but equally critical roles. Kinematically, the suspension allows the necessary vertical travel for the wheels to pass over obstacles, while simultaneously restricting unwanted longitudinal and lateral movements to ensure precise steering and stability. Dynamically, the suspension acts as an energy filter with two competing objectives. First, it must isolate the chassis from road vibrations to guarantee passenger comfort and protect the vehicle structure from fatigue. Second, it must maintain continuous contact between the tires and the road surface. A steady contact force is required to generate the lateral grip needed for safe cornering and to ensure effective braking [11].

Currently, passive suspension systems that use springs and viscous dampers with fixed coefficients remain the standard for commercial vehicles due to their mechanical simplicity, low manufacturing cost, and high reliability [7]. There are more advanced alternatives, including semi-active systems, which offer variable damping coefficients in real time through devices such as magneto-rheological dampers, and fully active systems, which employ power-controlled actuators that continuously adapt to changing conditions. Although these advanced systems offer improved ride quality and adaptability, their higher cost and complexity limit their widespread application [20].

Despite their mechanical simplicity, designing a passive system presents a complex engineering challenge due to the necessity of balancing conflicting objectives as the vehicle encounters random road excitations. Since the suspension parameters are

fixed, finding a configuration that performs well across all conditions is difficult. To address these difficulties, extensive research has been conducted to develop design frameworks based on mathematical optimization [15, 1, 18, 7, 30, 24]. The methodology developed in this thesis provides a systematic approach to identify the optimal suspension parameters in order to minimize driver discomfort.

1.2 Optimization in engineering systems and vehicle suspensions

In modern engineering, the goal is not only to develop functioning systems, but also to improve their performance through systematic procedures. Optimization plays a central role in this process as it searches for the set of design variables that minimize or maximize a defined performance metric, known as the objective function. A common classification of optimization algorithms distinguishes between gradient-free (stochastic) and gradient-based (deterministic) methods [4].

Stochastic methods rely on randomized search and mimic natural selection but often come with high computational cost. Despite this computational burden, Genetic Algorithms (GA) have been widely used in this field due to their ability to handle discrete variables and converge to the global minima when given a high number of function calls. In the domain of passive suspension, Mitra et al. [15] utilized these algorithms to optimize suspension and overall vehicle parameters specifically for ride comfort and road holding. The methodology extends to active systems, where researchers like Baumal et al. [1] and Mudduluru et al. [18] employed GAs to tune mechanical properties and PID controller gains, balancing the performance metrics with the limited working space of the suspension. More recently, Hu et al. [7] used the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to map the trade-offs between comfort and suspension deflection on a full-car model. Beyond GAs, other nature-inspired methods such as Particle Swarm Optimization (PSO) have also proven effective for better vibration isolation. [30]. In contrast, deterministic methods use the derivative of the objective function to efficiently guide the search more directly towards the optimal solution.

1.3 Gradient-based method and the Adjoint method

As the name suggests, gradient-based methods compute the derivative of the objective function F with respect to each of the design variables b_k, $k = 1, 2, \ldots, n$. These derivatives, also known as sensitivity derivatives (SDs), quantify how small changes in each design variable affect the objective function. The algorithm uses these sensitivities to iteratively update the design variables in a direction that minimizes or

maximizes F, depending on the objective.

Gradient-based methods can be further classified based on the level of derivative information they require:

- **First-order methods**: Examples include the Steepest Descent and Conjugate Gradient methods, which utilize only the first derivative (gradient) of the objective function.
- **Quasi-Newton methods**: Methods like BFGS approximate the second derivative (Hessian matrix) using information from the gradient, offering a balance between speed and computational cost.
- **Second-order methods**: Examples include Newton's method, which requires the computation of the exact Hessian matrix. While powerful, they are often computationally prohibitive for complex engineering problems.

The primary challenge in gradient-based optimization lies in how the gradient of the objective function is computed. The available methods differ in whether they calculate an approximate gradient or the exact analytical gradient, as well as in how their cost scales with the number of design variables and their implementation effort. For the sake of simplicity, we refer to the cost of solving the primal system (i.e., the system of the governing physical equations) once as one unit of computational cost.

Finite Differences (FD): The most common and simple approach to approximate the gradient is the Finite Difference method. The derivative is estimated by perturbing each design variable by an infinitesimally small step ϵ. The central difference (second-order) scheme to the k^{th} component of the gradient at $\mathbf{b}$ is:

$$\frac{\delta F}{\delta b_k} = \frac{F(b_1, \ldots, b_k + \epsilon, \ldots, b_n) - F(b_1, \ldots, b_k - \epsilon, \ldots, b_n)}{2\epsilon} \tag{1.1}$$

for $k = 1, 2, \ldots, n$. This method is easily applicable as it treats the solver as a "black box". It has been successfully applied in vehicle suspension optimization; for instance, Thoresson et al. [24] employed the Dynamic-Q optimization method using Central Finite Differences to mitigate the numerical noise that is inherent in vehicle dynamics simulations. However, FD has two major drawbacks:

1. **Computational Cost**: It requires at least n (for forward) or $2n$ (for central) additional solver evaluations per optimization cycle. For practical problems with many design variables, this cost is prohibitive.
2. **Accuracy/Truncation Error**: The accuracy heavily depends on the step size ϵ. If ϵ is too large, truncation error dominates. If too small, it runs the risk of either setting the ratio too high due to a very small denominator or having both numerator and denominator approximately zero.

Complex Variable method (CV): An alternative and improved method over FD is the Complex Variable method, which computes the derivative using a complex step perturbation $i\epsilon$:

$$\frac{\delta F}{\delta b_k} = \frac{\mathrm{Im}[F(b_k + i\epsilon)]}{\epsilon} \tag{1.2}$$

where $i = \sqrt{-1}$ and Im is the imaginary part of any complex variable. Unlike FD, the CV does not involve subtraction in the numerator, thus making it independent of the step-size sensitivity. However, it still requires n solver calls, meaning the cost still scales linearly with the number of design variables and the solver needs to be modified to support complex variables.

Direct Differentiation (DD): Direct Differentiation involves differentiating the primal system of equations with respect to the design variables. This results in n additional systems of linear equations (sensitivity equations) that must be solved. While DD provides the exact gradient (eliminating the approximation error of FD), its computational cost still scales with n, similar to the FD and CV methods.

Adjoint Method (AM): The Adjoint method offers an alternative, especially for problems with a large number of design variables that outnumber the objective functions. Its primary advantage is that it computes the exact gradient $\frac{\delta F}{\delta \mathbf{b}}$ at a cost that is independent of the number of design variables n. Once the primal equations are solved, the gradient is obtained by solving a single system of **adjoint equations**, followed by relatively inexpensive operations to evaluate the sensitivities. Thus, the cost is roughly equivalent to solving the primal problem just once more, regardless of the number of design variables. This efficiency makes it ideal for large-scale optimization.

1.4 Computational Challenges in Unsteady Adjoint

In unsteady (time-dependent) systems, the numerical solution of the adjoint equations necessitates access to the complete history of the state variables (i.e., the solution of the primal equations at each time-step). While storing this entire time series is manageable for low-dimensional problems, such as the full-vehicle model studied in this work, it becomes prohibitive for large-scale simulations encountered in Computational Fluid Dynamics (CFD) or Finite Element Analysis (FEA). In such cases, the memory requirements scale with the number of spatial nodes and time steps, often exceeding the available RAM. Since the objective of this thesis is to develop a robust adjoint optimization framework applicable to engineering dynamic systems, dealing with this storage limitation is a fundamental aspect of the

methodology, even if the specific vehicle model does not strictly impose it.

Recomputation: A straightforward approach to mitigate storage limits is recomputation, where the primal solution is re-calculated from the initial condition whenever a state is needed. While this minimizes memory usage, it leads to a substantial increase in computational cost (CPU time).

Approximation: A more viable alternative is approximation, where only selected time steps are stored, and the intermediate states are reconstructed using either simple interpolation methods (such as linear interpolation or cubic-splines) or Reduced Order Models (ROM), typically based on Proper Orthogonal Decomposition (POD) [29, 27, 28]. While efficient, these methods introduce approximation errors that can affect the accuracy of the computed gradients.

Check-pointing: The most frequently used strategy to balance memory and CPU cost is Check-pointing [5, 17]. This method strategically stores a subset of time-steps (checkpoints) during the primal solution. During the reverse adjoint integration, missing primal states are recomputed from the nearest checkpoint. The algorithm selects checkpoints to control the amount of total recomputation while respecting a user-defined memory limit.

More recently, to further address the memory bottleneck in large-scale adjoint problems, advanced lossy compression techniques have been developed. For instance, Margetis et al. [13] proposed a method in which the primal states are compressed using algorithms such as ZFP and incremental Proper Generalized Decomposition (iPGD). This strategy, referred to as *Compressed Full Storage*, significantly reduces the memory footprint to the point where the expensive recomputations of standard Check-pointing are avoided, while maintaining sufficient accuracy for the adjoint gradient computation.

1.5 Thesis Objectives and Layout

The primary objective of this diploma thesis is the development and implementation of an optimization framework for a full-vehicle suspension system to improve ride comfort, based on the unsteady continuous adjoint method. The goal is to compute the exact gradient of the objective function with respect to a vector of design variables at a computational cost independent of the number of variables, thereby enabling efficient gradient-based optimization.

The thesis is structured as follows:

- **Chapter 2** presents the mathematical modeling of the full-vehicle (8 degrees of freedom) and the governing equations of motion. It also describes the

modeling of road excitation profiles, and the numerical integration schemes used to solve the primal equations.

- **Chapter 3** is dedicated to the theoretical formulation of the continuous adjoint method. The adjoint equations, the boundary conditions, and the expression for the sensitivity derivatives are derived in detail.
- **Chapter 4** addresses the computational challenges associated with the unsteady adjoint method, specifically the memory requirements. It details the implementation of the **Check-pointing** algorithm, which manages the storage of primal states during the reverse adjoint integration to balance memory usage and computational cost.
- **Chapter 5** focuses on the application of the developed framework. Several optimization test cases are solved to demonstrate the effectiveness of the method. The results are presented and analyzed.
- Finally, **Chapter 6** summarizes the conclusions drawn from this study and provides suggestions for future research.

Chapter 2

The Vehicle Dynamics Model

2.1 Full-Vehicle Suspension Modeling

This study focuses on the vertical dynamic behavior of a passenger vehicle, neglecting lateral motion and steering effects. The system is modeled as a 7-degree-of-freedom (DoF) Full-Car (FC) suspension system coupled with a 1-DoF Driver's Seat (DS) model. This combined 8-DoF formulation has been widely utilized in the literature for suspension optimization [12, 16, 23, 2, 3]. The physical configuration of the system is illustrated in Figure 2.1.

2.1.1 Coordinate System

A right-handed inertial coordinate system is defined where the X-axis points longitudinally forward, the Y-axis points laterally to the left, and the Z-axis points vertically upwards.

Crucially, the origin of the vertical coordinate ($z = 0$) is defined to coincide with the *unloaded configuration* of the vehicle, where all suspension springs are at their free, uncompressed length. In this reference frame, the gravitational forces acting on the masses cause the vehicle to settle into a static equilibrium position at a non-zero vertical displacement ($z_{static} < 0$). Consequently, the state variables in the simulation represent the absolute position relative to this uncompressed frame, allowing for a direct evaluation of spring compression and suspension travel limitations. The origin of the coordinate system coincides with the Center of Gravity (CoG) of the chassis in its unloaded state. The front and rear axles are located at longitudinal distances a and b from the CoG, respectively, defining a constant wheelbase $l = a + b$. The left and right wheels are spaced symmetrically at a lateral

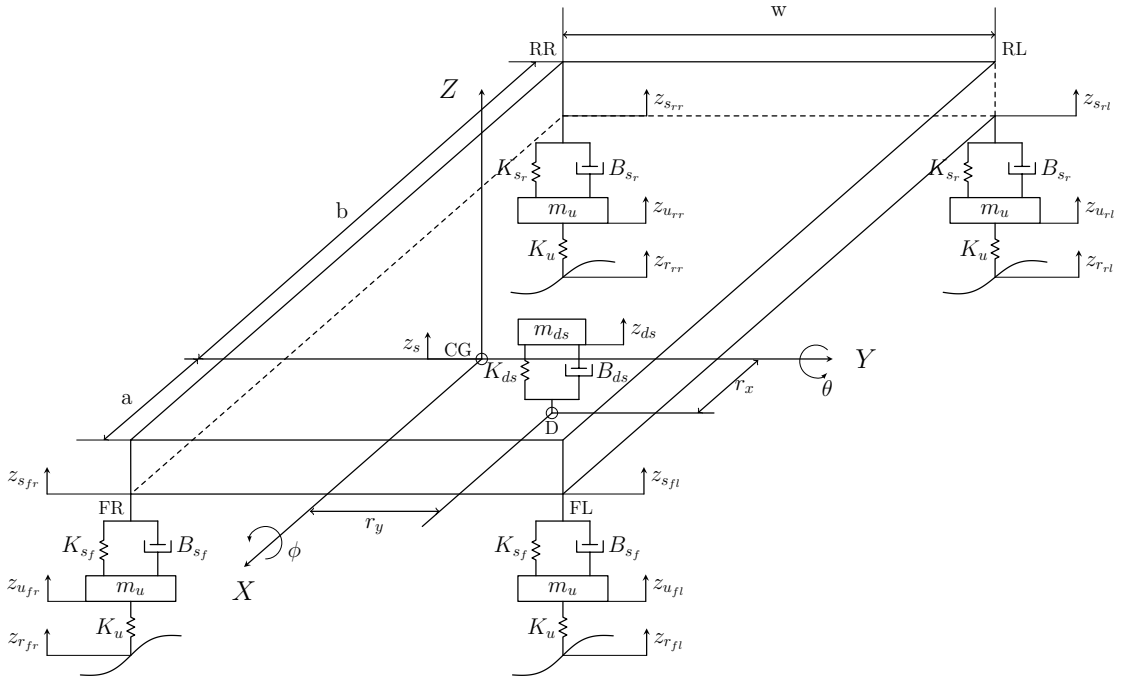


Figure 2.1: *Full-vehicle suspension model with driver seat.*

distance $\pm w/2$, where w is the track width of the vehicle. Consequently, while the CoG remains the fixed mathematical origin of the chassis, varying the parameter a during optimization shifts the relative placement of the axles. In terms of vehicle dynamics, this alters the static weight distribution and is practically described as shifting the CoG longitudinally along the vehicle's wheelbase.

2.1.2 Degrees of Freedom and State Variables

The dynamic behavior of the system is described by 8 independent degrees of freedom.

Chassis (Sprung Mass): The vehicle body (chassis) is modeled as a rigid sprung mass m_s. Its motion is described by three DoFs:

- **Heave (z_s):** The vertical translational displacement of the center of gravity (CoG).
- **Pitch (θ):** The rotational angle about the lateral Y-axis (nose-up/down motion).
- **Roll (ϕ):** The rotational angle about the longitudinal X-axis (leaning left/right).

While heave is measured in meters (m), pitch and roll are measured in radians (rad) as they describe rotational orientation. To visualize the distinction between the translational (heave) and rotational (pitch/roll) modes, a simplified representation

is provided in Figure 2.2.

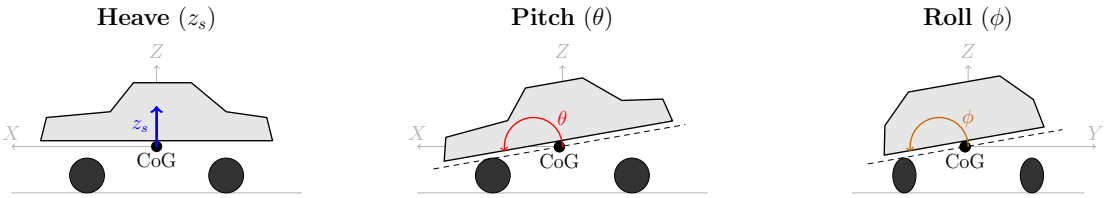


Figure 2.2: *Visual representation of the chassis degrees of freedom: Heave (vertical displacement), Pitch (longitudinal rotation) and Roll (lateral rotation).*

Wheels (Unsprung Masses): The chassis is connected to four wheels (unsprung masses, m_u) via suspension struts modeled as linear springs with stiffness K_s and viscous dampers with coefficient B_s. The vehicle is equipped with a fully independent suspension system, meaning that the vertical displacement of one wheel does not directly induce displacement in the opposite side wheel [11]. The model assumes lateral symmetry: the two front wheels share identical stiffness (K_{s_f}) and damping (B_{s_f}) coefficients, while the two rear wheels share their own distinct set of coefficients (K_{s_r}, B_{s_r}). Each wheel is modeled as a lumped unsprung mass m_u constrained to move only in the vertical direction. Therefore, there is one DoF (vertical displacement z_{u_i}) associated with each wheel. Since there are four wheels (Front-Left, Front-Right, Rear-Left, Rear-Right), this adds 4 DoFs to the system.

Driver's Seat: A dedicated Driver's Seat (DS) subsystem is included to evaluate ride comfort more accurately and adds 1 DoF (z_{ds}) to the system. This subsystem introduces the human comfort perception, since the driver experiences vibrations differently from the chassis frame. The seat is modeled as a secondary mass m_{ds} connected to the chassis by its own spring (K_{ds}) and damper (B_{ds}). The DS is positioned at a fixed longitudinal distance $r_x = 0.2\,m$ and a lateral distance $r_y = 0.375\,m$ from the CoG. The car is modeled as a left-hand drive vehicle (LHD), hence $r_x, r_y > 0$.

Table 2.1 summarizes the physical definitions, symbols, and units for all parameters used in the vehicle model. For the fixed parameters, single nominal values are provided. For the optimization design variables, the values are presented in the format $Lower\ Bound \leq Baseline \leq Upper\ Bound$, adopting the ranges established by Meng et al. [14] and Fossati et al. [3]. The baseline values characterize the initial configuration of the full-vehicle model prior to optimization. It should be noted that while a and b are both listed with their respective working bounds, they are geometrically coupled through the constant wheelbase constraint ($l = a+b = 2.6\,\text{m}$). Consequently, altering the front axle distance a automatically dictates the rear axle distance b.

Table 2.1: *Baseline Full-Vehicle Model Parameters (Prior to Optimization)*

Parameter	Symbol	Value	Unit
Suspension System			
Front Stiffness	K_{s_f}	$7{,}500 \leq 15{,}000 \leq 22{,}500$	N/m
Rear Stiffness	K_{s_r}	$8{,}500 \leq 17{,}000 \leq 25{,}500$	N/m
Front Damping	B_{s_f}	$1{,}250 \leq 2{,}500 \leq 3{,}750$	Ns/m
Rear Damping	B_{s_r}	$1{,}250 \leq 2{,}500 \leq 3{,}750$	Ns/m
Wheel mass (unsprung)	m_u	40	kg
Tire Stiffness	K_u	250,000	N/m
Driver Seat			
Seat position relative to CoG	r_x	0.2	m
Seat position relative to CoG	r_y	0.375	m
Seat Stiffness	K_{ds}	15,000	N/m
Seat Damping	B_{ds}	150	Ns/m
Driver and Seat mass	m_{ds}	100	kg
Chassis			
Car Wheelbase	l	2.6	m
Front Axle to CoG Distance	a	$1.04 \leq 1.2 \leq 1.43$	m
Rear Axle to CoG Distance	b	$1.17 \leq 1.4 \leq 1.56$	m
Car track width	w	1.5	m
Chassis Mass	m_s	1,100	kg
Roll Moment of Inertia	I_{xx}	550	kg·m^2
Pitch Moment of Inertia	I_{yy}	$1{,}500 \leq 1{,}848 \leq 2{,}100$	kg·m^2

2.1.3 Kinematics and Forces

The motion of the suspension mounting points (corners) depends on the rigid body motion of the chassis. Assuming small angles for pitch and roll ($\sin\theta \approx \theta, \cos\theta \approx 1$), the vertical displacement of any corner is a superposition of the heave, pitch, and roll contributions. Also, the vertical displacement of the point D that is where the driver's seat attaches to the chassis (Figure 2.1) is a superposition of heave, pitch and roll. For example, a positive pitch angle θ (nose down) lowers the front and raises the rear, while a positive roll angle (leaning right) lowers the right and raises

the left. The kinematic relationships for the four corners and the driver's seat are:

$$z_{s_{fl}} = z_s - a\theta + \frac{w}{2}\phi \tag{2.1a}$$

$$z_{s_{fr}} = z_s - a\theta - \frac{w}{2}\phi \tag{2.1b}$$

$$z_{s_{rl}} = z_s + b\theta + \frac{w}{2}\phi \tag{2.1c}$$

$$z_{s_{rr}} = z_s + b\theta - \frac{w}{2}\phi \tag{2.1d}$$

$$z_{s_{ds}} = z_s - r_x\theta + r_y\phi \tag{2.1e}$$

The suspension forces are generated by the relative motion and velocity between the chassis corners (z_{s_i}) and the wheels (z_{u_i}). The suspension force at the DS is generated by the relative motion and velocity between the DS (z_{ds}) and the point D ($z_{s_{ds}}$) from the point of attachment of the DS to the chassis. Modeling the suspension as a linear spring-damper system, the total force is the sum of the elastic force (Hooke's Law) and the viscous damping force. For the front-left corner and the driver's seat, this is expressed as:

$$F_{s_{fl}} = K_{s_f}(z_{s_{fl}} - z_{u_{fl}}) + B_{s_f}(\dot{z}_{s_{fl}} - \dot{z}_{u_{fl}}) \tag{2.2a}$$

$$F_{ds} = K_{ds}(z_{ds} - z_{s_{ds}}) + B_{ds}(\dot{z}_{ds} - \dot{z}_{s_{ds}}) \tag{2.2b}$$

Similar expressions apply to the other three corners ($F_{s_{fr}}, F_{s_{rl}}, F_{s_{rr}}$).

2.1.4 Equations of Motion

The governing differential equations are derived by applying Newton's Second Law to each mass in the system.

Sprung Mass (Chassis) Equilibrium: The vertical motion is governed by the balance of vertical forces. The mass times vertical acceleration equals the sum of forces from the four suspension struts, the driver's seat, and gravity:

$$m_s\ddot{z}_s = F_{ds} - W_s - (F_{s_{fl}} + F_{s_{fr}} + F_{s_{rl}} + F_{s_{rr}}) \tag{2.3}$$

For rotational equilibrium, the sum of moments about the CoG yields the angular accelerations. The pitch motion is driven by the longitudinal leverage of the front (a) and rear (b) axles:

$$I_{yy}\ddot{\theta} = -r_xF_{ds} + a(F_{s_{fl}} + F_{s_{fr}}) + b(-F_{s_{rl}} - F_{s_{rr}}) \tag{2.4}$$

Similarly, the roll motion is driven by the lateral leverage ($w/2$) of the left and right wheels:

$$I_{xx}\ddot{\phi} = r_y F_{ds} + \frac{w}{2}(-F_{s_{fl}} - F_{s_{rl}}) + \frac{w}{2}(F_{s_{fr}} + F_{s_{rr}}) \tag{2.5}$$

Driver's Seat Equilibrium: The vertical motion is governed by the balance of vertical forces. The mass times vertical acceleration equals the sum of forces from the driver's seat suspension and gravity:

$$m_{ds}\ddot{z}_{ds} = -F_{ds} - W_{ds} \tag{2.6}$$

Unsprung Masses (Wheels) Equilibrium: Each wheel is subjected to the suspension force from above and the tire reaction force from below. The tire is modeled as a linear spring with stiffness K_u, interacting with the road profile z_r. The equation of motion for the front-left wheel is:

$$m_u\ddot{z}_{u_{fl}} = F_{s_{fl}} - W_u - K_u(z_{u_{fl}} - z_{r_{fl}}) \tag{2.7}$$

The equations for the remaining wheels follow the same pattern.

2.1.5 State-Space Formulation

The resulting equations form a system of eight second-order Ordinary Differential Equations (ODEs) that describe the vehicle's motion. By introducing auxiliary variables, the system is converted into a first-order system consisting of sixteen coupled ODEs. This is crucial to facilitate numerical integration and the application of the adjoint method.

The state vector $\mathbf{x} \in \mathbb{R}^{16}$ comprises the displacements and velocities of the 8 DoFs, called **state variables**:

$$\mathbf{x} = [z_s, \dot{z}_s, \theta, \dot{\theta}, \phi, \dot{\phi}, \underbrace{z_{u_{fl}}, \dot{z}_{u_{fl}}, \ldots}_{\text{4 wheels}}, z_{ds}, \dot{z}_{ds}]^T \tag{2.8}$$

where:

$x_1 = z_s$	Chassis Heave Displacement
$x_2 = \dot{z_s}$	Chassis Heave velocity
$x_3 = \theta$	Chassis Pitch Angle
$x_4 = \dot{\theta}$	Chassis Pitch Angular Velocity
$x_5 = \phi$	Chassis Roll Angle
$x_6 = \dot{\phi}$	Chassis Roll angular velocity
$x_7 = z_{u_{fl}}$	Front-Left Wheel Vertical Displacement
$x_8 = \dot{z}_{u_{fl}}$	Front-Left Wheel Vertical Velocity
$x_9 = z_{u_{fr}}$	Front-Right Wheel Vertical Displacement
$x_{10} = \dot{z}_{u_{fr}}$	Front-Right Wheel Vertical Velocity
$x_{11} = z_{u_{rl}}$	Rear-Left Wheel Vertical Displacement
$x_{12} = \dot{z}_{u_{rl}}$	Rear-Left Wheel Vertical Velocity
$x_{13} = z_{u_{rr}}$	Rear-Right Wheel Vertical Displacement
$x_{14} = \dot{z}_{u_{rr}}$	Rear-Right Wheel Vertical Velocity
$x_{15} = z_{ds}$	Driver's Seat Vertical Displacement
$x_{16} = \dot{z}_{ds}$	Driver's Seat Vertical Velocity

The system is excited by road irregularities applied at the tire contact patches. These road disturbances are collected in the input vector $\mathbf{u}(t) \in \mathbb{R}^4$, which contains the time-varying road profile height at each wheel contact patch.

$$\mathbf{u}(t) = [z_{r_{fl}}(t), z_{r_{fr}}(t), z_{r_{rl}}(t), z_{r_{rr}}(t)]^T \tag{2.9}$$

where:

$u_1 = z_{r_{fl}}$	Terrain Disturbance Excitation at the Front-Left Wheel
$u_2 = z_{r_{fr}}$	Terrain Disturbance Excitation at the Front-Right Wheel
$u_3 = z_{r_{rl}}$	Terrain Disturbance Excitation at the Rear-Left Wheel
$u_4 = z_{r_{rr}}$	Terrain Disturbance Excitation at the Rear-Right Wheel

The **input variables** are external functions of time defined by the chosen road scenario (e.g., a deterministic bump or a stochastic road profile).

The **primal system** is linear and time-invariant (LTI) and is expressed in this final

state-space model form:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{S} \tag{2.10}$$

The source vector $\mathbf{S} \in \mathbb{R}^{16}$ contains the gravitational acceleration $-g$ terms acting on the vertical velocity states:

$$\mathbf{S} = \begin{bmatrix} 0 & -g & \mathbf{0}_4 & 0 & -g & \dots & 0 & -g \end{bmatrix}^T \tag{2.11}$$

The system matrix $\mathbf{A} \in \mathbb{R}^{16\times16}$ is partitioned into sub-matrices representing the coupling between the Chassis (C), Wheels (W), and Driver's Seat (D):

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{CC} & \mathbf{A}_{CW} & \mathbf{A}_{CD} \\ \mathbf{A}_{WC} & \mathbf{A}_{WW} & \mathbf{0}_{8\times2} \\ \mathbf{A}_{DC} & \mathbf{0}_{2\times8} & \mathbf{A}_{DD} \end{bmatrix} \tag{2.12}$$

Here, $\mathbf{A}_{CC} \in \mathbb{R}^{6\times6}$ describes the chassis dynamics, $\mathbf{A}_{WW} \in \mathbb{R}^{8\times8}$ describes the wheel dynamics, and the off-diagonal blocks $\mathbf{A}_{CW}$, $\mathbf{A}_{WC}$ represent the suspension forces coupling the chassis and the wheels. Also, $\mathbf{A}_{DD} \in \mathbb{R}^{2\times2}$ describes the driver's seat dynamics, and the off-diagonal blocks $\mathbf{A}_{DC}$ and $\mathbf{A}_{CD}$ represent the the seat's suspension forces coupling the chassis and the driver's seat.

Finally, the input matrix $\mathbf{B} \in \mathbb{R}^{16\times4}$ maps the road displacements to the wheel vertical velocities:

$$\mathbf{B} = \left[\begin{array}{c|cccccccc|c} & 0 & \frac{K_u}{m_u} & 0 & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & \frac{K_u}{m_u} & 0 & 0 & 0 & 0 & \\ \mathbf{0}_{4\times6} & 0 & 0 & 0 & 0 & 0 & \frac{K_u}{m_u} & 0 & 0 & \mathbf{0}_{4\times2} \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{K_u}{m_u} & \end{array}\right]^T \tag{2.13}$$

2.1.6 Modeling Limitations

In this paper, the adopted formulation is based on several simplifying assumptions that lead to important physical limitations. These limitations do not prevent the model from being useful for ride analysis, but they must be respected when choosing road excitations and interpreting the results.

Linear tire model and wheel hop. To maintain contact between the tire and the road, the vertical displacement of the wheel z_u must be less than or equal to the road excitation z_r. When $z_u > z_r$, the tire loses contact with the road and the normal force drops to zero. The phenomenon of wheel hop or tyre hop introduces nonlinearities that are known to significantly alter the vehicle's dynamic behavior [26]. Thoresson

et al. [24] observed that when optimizing for ride comfort, the design variables related to damping exhibit a sensitivity to wheel hop. Physically, a tire can only exert a compressive normal force on the road surface so a physically consistent tire force model is:

$$F_{tire} = \begin{cases} -K_u(z_u - z_r), & \text{if } z_u \leq z_r \ (contact/compession) \\ 0, & \text{if } z_u > z_r \ (lift - off) \end{cases}$$

However, enforcing this one-sided constraint makes the system piecewise linear and non-differentiable at the switching point. Implementing this nonlinearity would cause the stiffness terms in the system matrix **A** to vary depending on the state of the vehicle (specifically, the relative displacement $z_u - z_r$). This would render the system *nonlinear* and *state-dependent*. Since this work relies on a **Linear Time-Invariant (LTI)** formulation where the system matrix **A** must remain constant, the tire is modeled as a linear spring:

$$F_{\text{tire}} = -K_u(z_u - z_r)$$

which can produce non-physical tensile forces when $z_u > z_r$ (wheel hop). To ensure the validity of the model, the amplitudes and frequencies of the road excitations are selected such that there is no separation between the tires and the road.

Suspension working space. All suspension springs and dampers are modeled as linear elements, and no mechanical travel limits (bump or rebound stops) are included. Consequently, the model does not prevent excessive suspension deflection under large excitations that may cause the suspension to hit the upper and lower stops. In some studies, they enforce suspension working space and road-holding by implementing them as inequality constraints that penalize the objective function when breached[1, 23, 18]. Thoresson et al. [24] in his study does not add a constraint for suspension travel, as their model accounted for physical limits though the inclusion of non-linear bump and rebound stops. Nonetheless, neglecting mechanical stops is acceptable for moderate road conditions, where suspension travel remains within normal limits. Under severe off-road excitations or large suspension strokes, however, this assumption becomes invalid because travel limits introduce nonlinear effects.

Small-angle assumption. The pitch and roll angles of the vehicle are assumed to remain small, allowing the car model to retain its linear nature. $\sin\theta \approx \theta$, $\cos\theta \approx 1$, $\sin\phi \approx \phi$, and $\cos\phi \approx 1$. Accordingly, the model is accurate for operating regimes with small body rotations, but its fidelity decreases for terrains that induce large pitch/roll amplitudes.

Road Inclination. The gravitational source vector **S** is modeled as a constant vertical force aligned with the inertial Z-axis, which implies the vehicle operates on a horizontal global plane. While the model is fully capable of simulating local unevenness—such as asymmetric bumps or potholes applied to individual wheels—it does not account for the change in static load of the that occurs on an incline. Specifically, the model continues to apply the full gravitational load (g) to the suspension even on slopes, ignoring the fact that the normal force supporting the vehicle should effectively decrease (to $g\cos\gamma$) as the road slope γ increases.

2.2 Road Excitation Modeling

The interaction between the road and the vehicle occurs through the vertical displacements at the four wheel contact points, denoted by $z_{r_{fl}}(t)$, $z_{r_{fr}}(t)$, $z_{r_{rl}}(t)$ and $z_{r_{rr}}(t)$. These road disturbances are defined by the specific terrain profile under investigation. In reality, a road surface contains both large, isolated irregularities—such as potholes, speed bumps, and patches—and continuous, smaller irregularities, like asphalt texture, small stones, and ripples. In the literature [10, 25, 3], this duality is commonly modeled by employing stochastic methods (typically a Gaussian random process with a Power-Law PSD) to characterize the small-scale pavement roughness, while using deterministic shapes to represent discrete events such as bumps, dips, potholes, or ramps. To provide a comprehensive evaluation of the Full-Vehicle suspension system this work employs a wide range of excitation tests while strictly respecting the modeling limitations.

Regardless of the nature of the excitation, it must ensure that the rear axle encounters the road disturbance at the exact spatial location previously encountered by the front axle. That means that if the excitation from the front wheels starts at $t = T_{start}$, the rear wheels experience the same profile after a time delay τ. Consequently, the rear wheels travel on a flat surface until $t = T_{start} + \tau$. The time delay τ is defined by the wheelbase $l = a + b$ and the vehicle longitudinal speed v as $\tau = \frac{a+b}{v}$. The disturbances are applied over a finite time interval $T_{start} \leq t \leq T_{end}$, where T_{start} is the activation time of the disturbance, and T_{end} is the final simulation time.

2.2.1 Deterministic Road Irregularities

The first category of excitation focuses on the vehicle's response to isolated irregularities. These scenarios are critical for analyzing the system's response amplitudes, settling time and stability following a disturbance. A smooth cosine bump is imple-

mented:

$$z_r(t) = \begin{cases} \frac{H}{2}\left(1 - \cos\left(\frac{2\pi v(t-T_{start})}{L_{bump}}\right)\right) & \text{for } T_{start} \leq t \leq T_{start} + \frac{L_{bump}}{v} \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

Various scenarios can be investigated; for example, a symmetric bump ($H > 0$) applied simultaneously to both left and right wheels in order to test heave and pitch damping. Alternatively, an asymmetric pothole ($H < 0$) applied only to the right wheels can be used to test roll stability. In the specific case presented below, the four wheels are subjected to different bump amplitudes: $H_{\text{left}} = 0.05\,\text{m}$ and $H_{\text{right}} = 0.06\,\text{m}$, with a bump length of $L_{\text{bump}} = 2.5\,\text{m}$ and a vehicle velocity of $v = 10\,\text{m/s}$. Also the vehicle's front wheels encounter the obstacle at $T_{start} = 0\,s$.

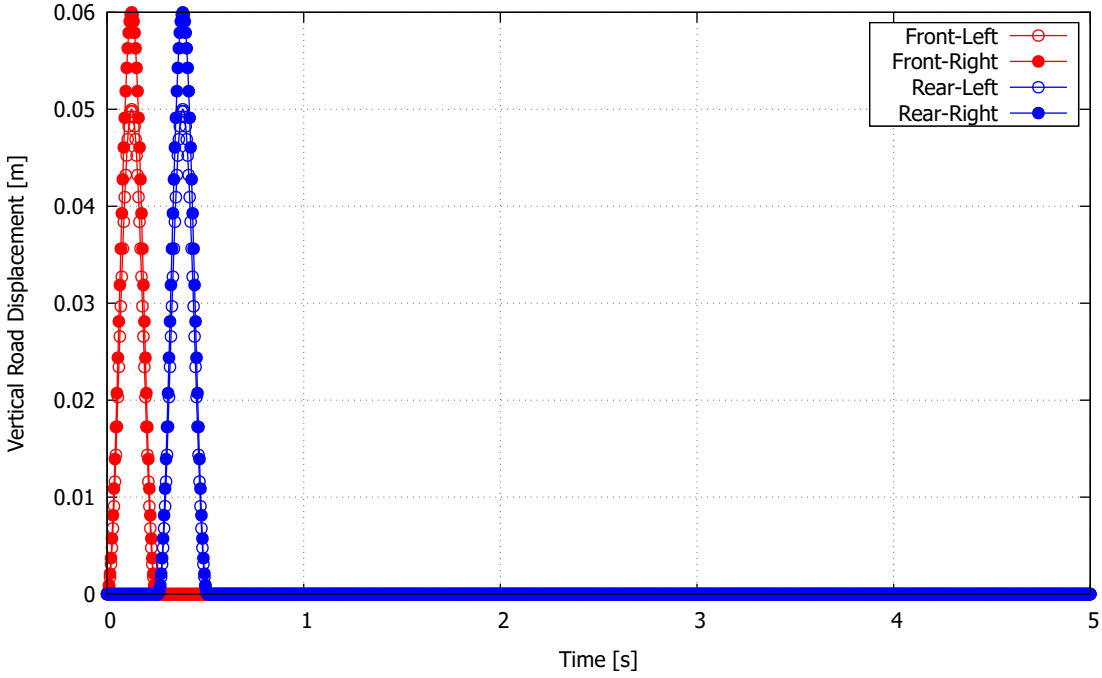


Figure 2.3: *Time-domain of Deterministic road profile excitation: asymmetric inputs, $v = 10\,m/s$*

2.2.2 Frequency Response to Harmonic Excitation

Another important category utilizes continuous periodic excitations to evaluate the system's frequency response. Subjecting the vehicle to continuous forcing allows the dynamic behavior to be analyzed independent of initial transient effects. The system exhibits a transient response that decays over time, but eventually the vibrations settle into a steady-state response. This final state is characterized by oscillations at the excitation frequency but with a different amplitude and phase angle. Analyzing

this steady state is crucial for identifying dynamic properties like resonance and system damping properties. A continuous sinusoidal excitation is assumed for each wheel, described by:

$$
\begin{aligned}
z_{r_{fl}} &= \begin{cases} A_{fl}\sin(\omega t), & \text{for } t \in [T_{start}, T_{end}] \\ 0, & \text{otherwise} \end{cases} \\
z_{r_{fr}} &= \begin{cases} A_{fr}\sin(\omega t), & \text{for } t \in [T_{start}, T_{end}] \\ 0, & \text{otherwise} \end{cases} \\
z_{r_{rl}} &= \begin{cases} A_{rl}\sin(\omega (t-\tau)), & \text{for } t \in [T_{start}+\tau, T_{end}] \\ 0, & \text{otherwise} \end{cases} \\
z_{r_{rr}} &= \begin{cases} A_{rr}\sin(\omega (t-\tau)), & \text{for } t \in [T_{start}+\tau, T_{end}] \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
\tag{2.15}
$$

where ω is the terrain excitation frequency, and $A_{fl}, A_{fr}, A_{rl}, A_{rr}$ are the amplitudes of the road disturbance on the respective wheels. In the specific case presented below, the four wheels are subjected to the following harmonic excitation: $\omega = 15\,rad/s$ and $A_{fl} = A_{rl} = 0.05\,m$, $A_{fr} = A_{rr} = 0.05\,m$ and a vehicle velocity of $v = 12\,\mathrm{m/s}$.

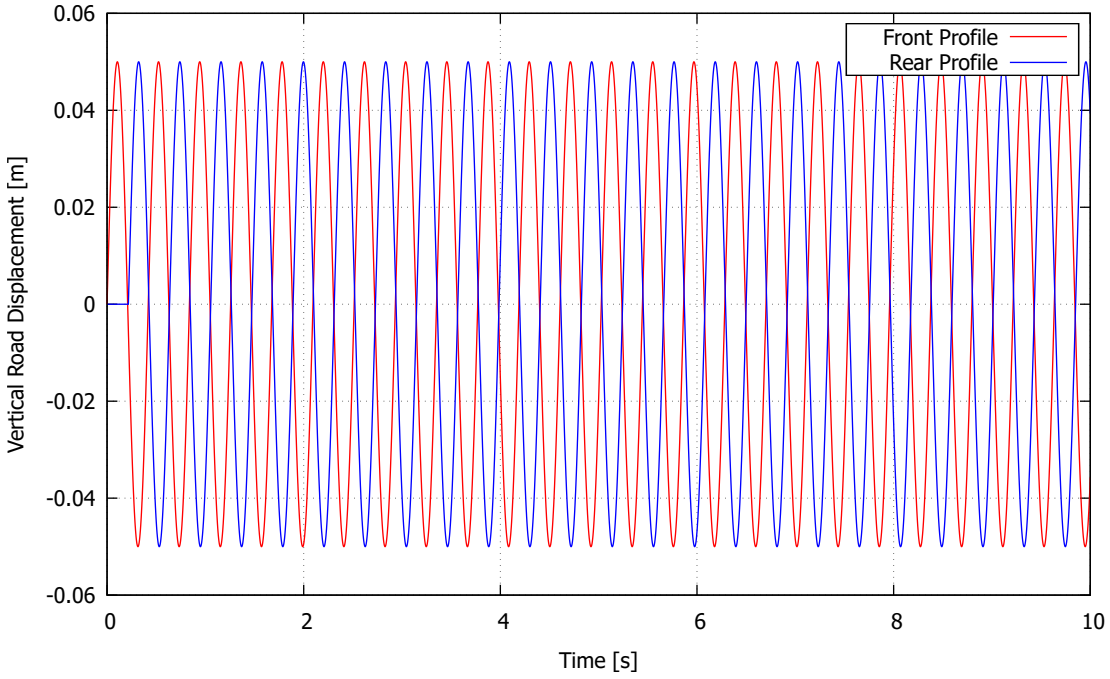


Figure 2.4: *Time domain of Harmonic road profile excitation: symmetric inputs, $v = 12\,m/s$*

2.2.3 Stochastic Road Profiles

To replicate realistic driving conditions, a stochastic road profile is generated via harmonic superpositions based on the ISO 8608 standard [10]. This standard classi-

Table 2.2: *ISO classes of road surface roughness $G_d(n_0)$ profiles ($n_0 = 0.1\,\mathrm{cycle/m}$)*

Road Class Category	**Geometric Mean** ($10^{-6}\mathrm{m}^3$)	**Range**
A (Very Good)	16	< 32
B (Good)	64	32 – 128
C (Average)	256	128 – 512
D (Poor)	1024	512 – 2048
E (Very Poor)	4096	2048 – 8192

fies road roughness using a Power Spectral Density (PSD) function, $G_d(n)$, expressed as a function of the spatial frequency n (cycles/m). The PSD follows a power-law relationship described by:

$$G_d(n) = G_d(n_0) \cdot \left(\frac{n}{n_0}\right)^{-w} \tag{2.16}$$

where $n_0 = 0.1\,\mathrm{cycles/m}$ serves as the reference spatial frequency and $w = 2$ represents the waviness coefficient typically associated with road surfaces exhibiting a constant velocity PSD. The Table 2.2 shows the different roughness values of class roads.

To generate a time-domain signal $z_r(t)$ suitable for the numerical solver, the spatial frequency profile is transformed using the vehicle's longitudinal speed v. The profile is constructed using the superposition of N discrete harmonics, as detailed by Tyan et al. [25]:

$$z_r(t) = \sum_{k=1}^{N} \sqrt{2G_d(n_k)\Delta n} \cdot \sin(2\pi n_k v t + \phi_k) \tag{2.17}$$

The spectrum is bounded by n_{min} and n_{max}, with the frequency step defined as $\Delta n = (n_{max} - n_{min})/N$. The discrete spatial frequencies are given by $n_k = n_{min} + (k-1)\Delta n$. The vehicle longitudinal speed v transforms these spatial components into temporal frequencies $\omega_k = 2\pi n_k v$. To ensure the simulation accurately captures the road surface, the time step Δt must be carefully selected. As the vehicle speed v increases, the wheels encounter road irregularities more frequently. If the time step is too large, the simulation might "skip over" the finer texture of the road (high-frequency components), causing the solver to miss key disturbances. To prevent this aliasing

and ensure the numerical solver sees the full roughness of the profile, the sampling rate is chosen to satisfy the standard Nyquist-Shannon criterion [22]:

$$\frac{1}{\Delta t} > 2 \cdot n_{max} \cdot v \tag{2.18}$$

where $1/\Delta t$ represents the sampling rate. The stochastic nature of the profile is ensured by independent random phase angles ϕ_k uniformly distributed in $[0, 2\pi]$. Although the Adjoint Method with Check-pointing supports extended simulation time bounds, the computational cost of re-integration motivates the use of representative short-duration segments. The stochastic road profile is generated using a fixed random seed in order to remain the same through every optimization cycle.

In the specific case presented below, the vehicle is subjected to a Class A (Very Good Asphalt) road profile with the roughness coefficient at $G_d(n_0) = 16 \times 10^{-6}\,\mathrm{m}^3$. This profile was generated for the current analysis using $N = 500$ harmonics with spatial frequencies ranging from $n_{\min} = 0.01\,\mathrm{cycles/m}$ to $n_{\max} = 3\,\mathrm{cycles/m}$ and a vehicle velocity of $v = 15\,\mathrm{m/s}$. The resulting profile is applied simultaneously to the left and right wheels.

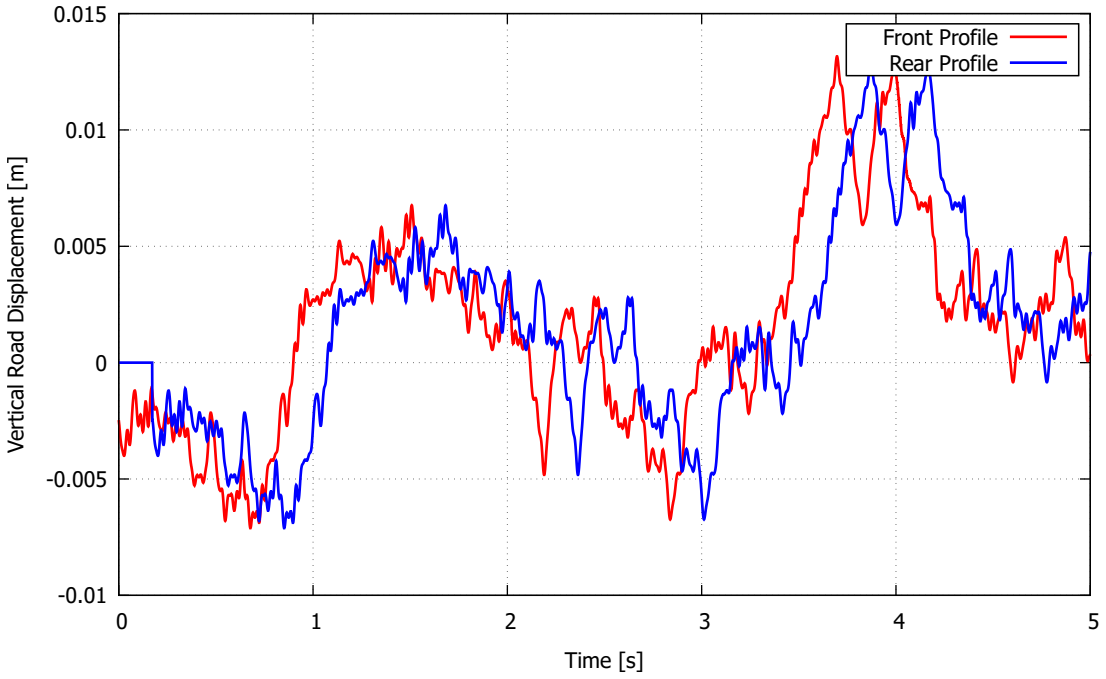


Figure 2.5: *Time-domain of the ISO 8608 Class A stochastic road profile: symmetric inputs, $v = 15\,m/s$*

2.3 Numerical Integration of Primal Equations

2.3.1 Steady-state solver: Gauss Elimination with Partial Pivoting

In order to solve the primal system 2.10, as with any system of ODEs, initial conditions for the state variables are required. These are obtained by allowing the system/vehicle to reach a static equilibrium under its own weight, before the application of the dynamic road disturbances. For this purpose, the unsteady primal system is transformed into its steady-state form by setting all time derivatives to zero. The resulting system is reformulated as a linear algebraic equation:

$$A\mathbf{x}_{ss} = -\mathbf{S} \tag{2.19}$$

where:

$\mathbf{x_{ss}} \in \mathbb{R}^{16\times 1}$ is the unknown vector of steady-state variables

This system is solved using the Gaussian Elimination method with Partial Pivoting, which is a direct method that consists of two primary phases:

Forward Elimination & Partial Pivoting

The goal is to transform the matrix A into an upper triangular form. Consider the system of equations in the following form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= -s_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= -s_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= -s_n \end{aligned} \tag{2.20}$$

In the first step, the first unknown x_1 is eliminated from all equations below the first row. This is accomplished by selecting the first equation as the pivot equation and using it to eliminate x_1 from subsequent rows. Specifically, the first equation is divided by its leading coefficient a_{11} and then multiplied by the corresponding coefficient $a_{21}, a_{31}, \ldots, a_{n1}$ of the rows below. These scaled versions are subtracted from the respective rows to eliminate x_1, thus completing the first step.

This process is then repeated: the second equation becomes the new pivot equation and its leading coefficient a'_{22} is used to eliminate x_2 from the rows below. After a

total of $n-1$ steps, the matrix is transformed into an upper triangular form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= -s_1 \\ a'_{22}x_2 + a'_{23}x_3 + \ldots + a'_{2n}x_n &= -s'_2 \\ a''_{33}x_3 + \ldots + a''_{3n}x_n &= -s''_3 \\ &\vdots \\ a^{(n-1)}_{nn}x_n &= -s^{(n-1)}_n \end{aligned} \tag{2.21}$$

To improve numerical stability and reduce truncation (round-off) errors, partial pivoting is introduced. At each elimination step, the algorithm searches the current column from the diagonal element downward and selects the element with the largest absolute value. The current row is then swapped with the row containing this maximum value, which is the new pivot element. This ensures that the pivot used for elimination is as great as possible in magnitude, which mitigates division by small numbers, and thus the risk of numerical instability.

Back substitution

Once the matrix has been transformed into the upper triangular form through forward elimination, the unknowns can be computed sequentially starting from the last equation, which contains only one unknown:

$$x_n = \frac{b^{(n-1)}_n}{a^{(n-1)}_{nn}} = \frac{b'_n}{a'_{nn}} \tag{2.22}$$

Then, the second-to-last equation contains two unknowns, but since x_n is already known, it reduces the $(n-1)^{th}$ equation also to one unknown x_{n-1} This procedure is repeated for each row moving upward, until all unknowns have been computed by the formula:

$$x_i = \frac{1}{a'_{ii}} \left[b'_i - \sum_{j=i+1}^{n} a'_{ij}x_j \right] \quad \text{for } i = n-1, \ldots, 1 \tag{2.23}$$

The solution produced by the algorithm is a row vector x_{ss} of the state variables that will be used as the initial condition for the unsteady primal problem that follows. The procedure can be summarized as Algorithm 1

Algorithm 1 Steady-State Solver: Gauss Elimination with Partial Pivoting

Require: System Matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, RHS vector $\mathbf{B}_{\text{rhs}} \in \mathbb{R}^n$
Ensure: Steady-state solution $\mathbf{x}_{\text{ss}} \in \mathbb{R}^n$

Phase 1: Forward Elimination

for $k = 1$ to $n - 1$ **do**
  Partial Pivoting:
  Find row p ($k \leq p \leq n$) with maximum $|A_{pk}|$
  if $p \neq k$ **then**
    Swap row k and row p in $\mathbf{A}$
    Swap element k and element p in $\mathbf{B}_{\text{rhs}}$
  end if
  Elimination:
  for $i = k + 1$ to n **do**
    multiplier $m \leftarrow A_{ik}/A_{kk}$
    $A_{ik} \leftarrow 0$
    for $j = k + 1$ to n **do**
      $A_{ij} \leftarrow A_{ij} - m \cdot A_{kj}$
    end for
    $B_i \leftarrow B_i - m \cdot B_k$
  end for
end for

Phase 2: Back Substitution

Initialize $\mathbf{x}_{\text{ss}}$
$x_n \leftarrow b_n/A_{nn}$
for $i = n - 1$ down to 1 **do**
  sum $\leftarrow B_i$
  for $j = i + 1$ to n **do**
    sum $\leftarrow$ sum $- A_{ij} \cdot x_j$
  end for
  $x_i \leftarrow$ sum$/A_{ii}$
end for
return $\mathbf{x}_{\text{ss}}$

2.3.2 Time integration of the unsteady primal problem: Runge-Kutta 4th order solver

The primal problem is an initial value problem, defined as a system of first-order ODEs and can be represented as

$$\frac{dx}{dt} = f(t, x(t)) \quad , \quad x(t_0) = x_0 \tag{2.24}$$

where f is smooth (guaranteed by the LTI formulation and the use of continuous, differentiable input functions), $x(t)$ is the real solution at t and x will express the unknown solution that should be approximated numerically over a time interval $t \in [t_{start}, t_{end}]$. Numerical methods for solving such problems can be classified into one-step and multi-step methods. One-step methods compute the next value x_{i+1} using only information from the current step, specifically the interval $[t_i, t_{i+1}]$. This makes them simple to implement and particularly suitable for variable step-size integration. In contrast, multi-step methods require information from multiple previous time steps. Although this often improves efficiency and reduces the number of function evaluations per step, it also introduces additional complexity. In this work, the time integration of the unsteady primal problem is carried out using the classical fourth-order Runge-Kutta method (RK4) with a fixed timestep Δt. This method belongs to the category of one-step methods and achieves fourth-order accuracy, with a local truncation error of $\mathcal{O}(\Delta t^5)$. The underlying idea of a 4^{th} Runge-Kutta method is to replace the Taylor series expansions

$$\begin{aligned} x(t_{i+1}) = x(t_i + \Delta t) &= x(t_i) + \sum_{n=1}^{N} \frac{\Delta t^n}{n!} x^{(n)}(t_i) + \mathcal{O}(\Delta t^{N+1}) \\ &= x(t_i) + \Delta t\, f(t_i, x(t_i)) + \frac{\Delta t^2}{2!}\, \dot{f}(t_i, x(t_i)) \\ &\quad + \frac{\Delta t^3}{3!}\, \ddot{f}(t_i, x(t_i)) + \frac{\Delta t^4}{4!}\, \dddot{f}(t_i, x(t_i)) + \mathcal{O}(\Delta t^5) \end{aligned} \tag{2.25}$$

where

$$\begin{aligned} \ddot{x}(t_i) = \frac{d}{dt}\left(\frac{dx(t_i)}{dt}\right) = \frac{d}{dt} f(t_i, x(t_i)) &= \frac{\partial f(t_i, x(t_i))}{\partial t} + \frac{\partial f(t_i, x(t_i))}{\partial t} \frac{dx(t_i)}{dt} \\ &= \frac{\partial f(t_i, x(t_i))}{\partial t} + \frac{\partial f(t_i, x(t_i))}{\partial t} f(t_i, x(t_i)) \end{aligned} \tag{2.26}$$

with an equation of the following form

$$x(t_{i+1}) = x(t_i + \Delta t) = x(t_i) + \sum_{i=1}^{4} w_i k_i \tag{2.27}$$

The RK4 method evaluates the right-hand side function $f(t, x)$ at four distinct points within each time step.

$$\begin{aligned} k_1 &= \Delta t\ f(t_i, x_i) \\ k_2 &= \Delta t\ f(t_i + a_1 \Delta t, x_i + b_{11} k_1) \\ k_3 &= \Delta t\ f(t_i + a_2 \Delta t, x_i + b_{11} k_2) \\ k_4 &= \Delta t\ f(t_i + a_1 \Delta t, x_i + b_{11} k_3) \end{aligned} \tag{2.28}$$

These evaluations are then combined through a weighted average to approximate the solution at the next time step. A typical scheme of the RK4 method is

$$x_{i+1} = x_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.29}$$

where

$$\begin{aligned} k1 &= \Delta t\ f(t_i, x_i) \\ k2 &= \Delta t\ f(t_i + \frac{\Delta t}{2}, x_i + \frac{k_1}{2}) \\ k3 &= \Delta t\ f(t_i + \frac{\Delta t}{2}, x_i + \frac{k_2}{2}) \\ k4 &= \Delta t\ f(t_{i+1}, x_i + k_3) \end{aligned} \tag{2.30}$$

and can be summarized as Algorithm 2 In summary, the steady-state solution x_{ss} provides the initial condition for the Runge-Kutta time integration (Algorithm 2) which generates the time history of the state variables for the unsteady primal system. This solution $\mathbf{x}$(t) serves as the basis for the adjoint formulation, where sensitivities with respect to the design variables are derived.

Algorithm 2 Unsteady Primal Solver: Runge-Kutta 4th Order

Require: Initial state $\mathbf{x}_0$, Matrices $(\mathbf{A}, \mathbf{B}, \mathbf{S})$, Road Profile $\mathbf{u}(t)$
Require: Time settings: t_{start}, t_{end}, step Δt
Ensure: Time history of states $\mathbf{x}(t)$
Initialize $\mathbf{x}(t_{\text{start}}) \leftarrow \mathbf{x}_0$
$N_{\text{steps}} \leftarrow (t_{\text{end}} - t_{\text{start}})/\Delta t$
for $i = 0$ to $N_{\text{steps}} - 1$ **do**
    $t_i \leftarrow t_{\text{start}} + i \cdot \Delta t$
    System Dynamics: $\mathbf{f}(t, \mathbf{x}(t)) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{S}$
    Compute Slopes:
    $\mathbf{k}_1 \leftarrow \Delta t \cdot \mathbf{f}(t_i, \quad \mathbf{x}_i)$
    $\mathbf{k}_2 \leftarrow \Delta t \cdot \mathbf{f}(t_i + \frac{\Delta t}{2}, \quad \mathbf{x}_i + \frac{1}{2}\mathbf{k}_1)$
    $\mathbf{k}_3 \leftarrow \Delta t \cdot \mathbf{f}(t_i + \frac{\Delta t}{2}, \quad \mathbf{x}_i + \frac{1}{2}\mathbf{k}_2)$
    $\mathbf{k}_4 \leftarrow \Delta t \cdot \mathbf{f}(t_i + \Delta t, \quad \mathbf{x}_i + \mathbf{k}_3)$
    Update State:
    $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$
end for

2.3.3 Numerical Demonstration

To demonstrate the performance of the steady-state and unsteady solvers, the full vehicle model (baseline configuration) was simulated traveling over the single deterministic road irregularity test case of Figure 2.3 in Subsection 2.2.1. The vehicle encounters the obstacle at $T_{start} = 0\,s$ and the simulation ends at $T_{end} = 5\,s$. The simulation is initialized using the static equilibrium vector x_{ss} computed by the Gaussian Elimination Algorithm 1 (Subsection 2.3.1), so the vehicle starts in a relaxed state. Subsequently, the Runge Kutta Algorithm 2 integrates the equations forward in time using the initial condition $x(0) = x_{ss}$.

Figure 2.6 illustrates the time evolution of four key degrees of freedom: Chassis heave displacement (z_s), pitch angle θ and roll angle ϕ, and Driver's Seat displacement (z_{ds}). The results highlight the consistency of the numerical framework. After the disturbance passes, the system dissipates the induced energy and converges back to the initial steady-state value, which is the same that the steady-state solver predicted. It is apparent that the steady-state of the pitch and roll variables is non-zero. This behavior arises from the eccentric position of the driver's mass ($r_x, r_y > 0$) relative to the CoG, combined with the fact that the vehicle's CoG rarely coincides with the exact geometric center of the track width and wheelbase. These offset masses generate static moments around the pitch and roll axes, causing the chassis to tilt slightly to achieve static equilibrium. Validating the physical modeling assumptions is also mandatory to ensure the integrity of the results. As noted in Subsection 2.1.6, the tire is modeled as a linear spring, a formulation that is physically valid only if the tire remains in contact with the road ($z_u \leq z_r$). If the wheel were to lift off

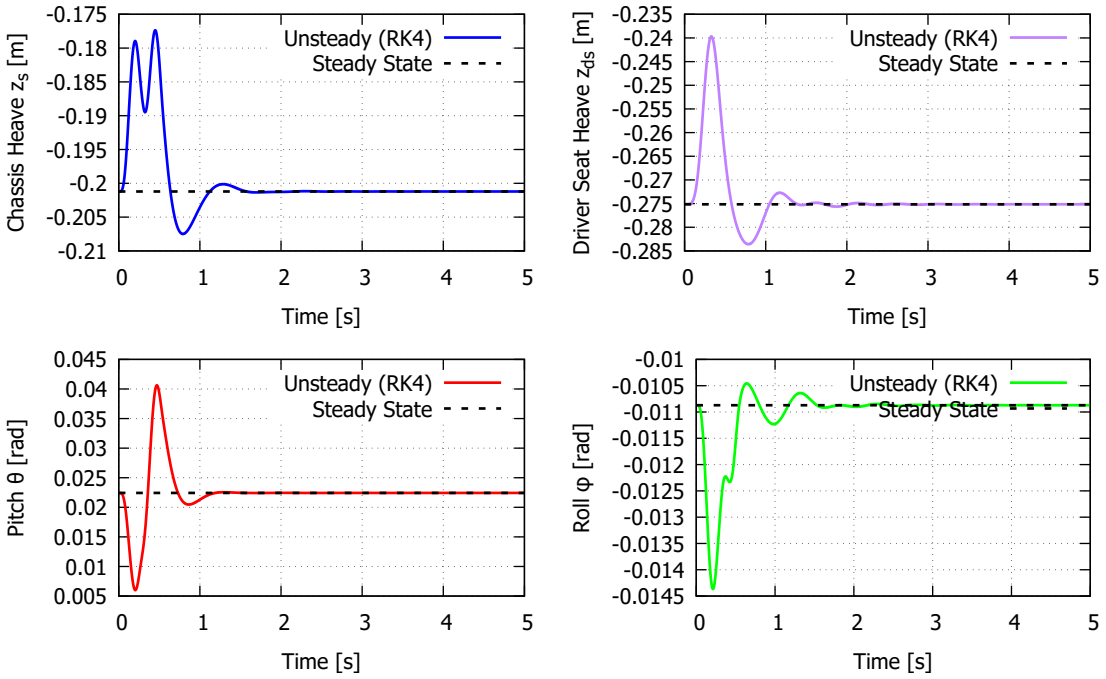


Figure 2.6: *System response to the deterministic bump excitation. The unsteady solution (solid lines) starts at the static equilibrium calculated by the Gauss solver (dashed lines), undergoes transient motion, and asymptotically returns to the same equilibrium.*

($z_u > z_r$), the linear model would incorrectly generate tensile forces.

Figure 2.7 plots the time history of the tire deflection, defined as $(z_r - z_u)$, for all four wheels. The results demonstrate that the deflection remains positive throughout the event. This confirms that the wheels maintain continuous contact with the road surface and no wheel hop phenomena occur. Finally, the validity of the infinite suspension working space assumption was examined. The current model neglects mechanical stops (bump and rebound stops), assuming that the suspension travel remains within the physical limits of the damper stroke.

Figure 2.8 displays the relative displacement between the chassis corners and the wheels, defined as $(z_{s_i} - z_{u_i})$. The maximum peak-to-peak travel observed is well within the typical working range of commercial vehicle suspensions (approximately ± 10 cm). This confirms that for the investigated maneuver, the simplifying assumption of neglecting mechanical bump stops does not compromise the validity of the dynamic analysis.

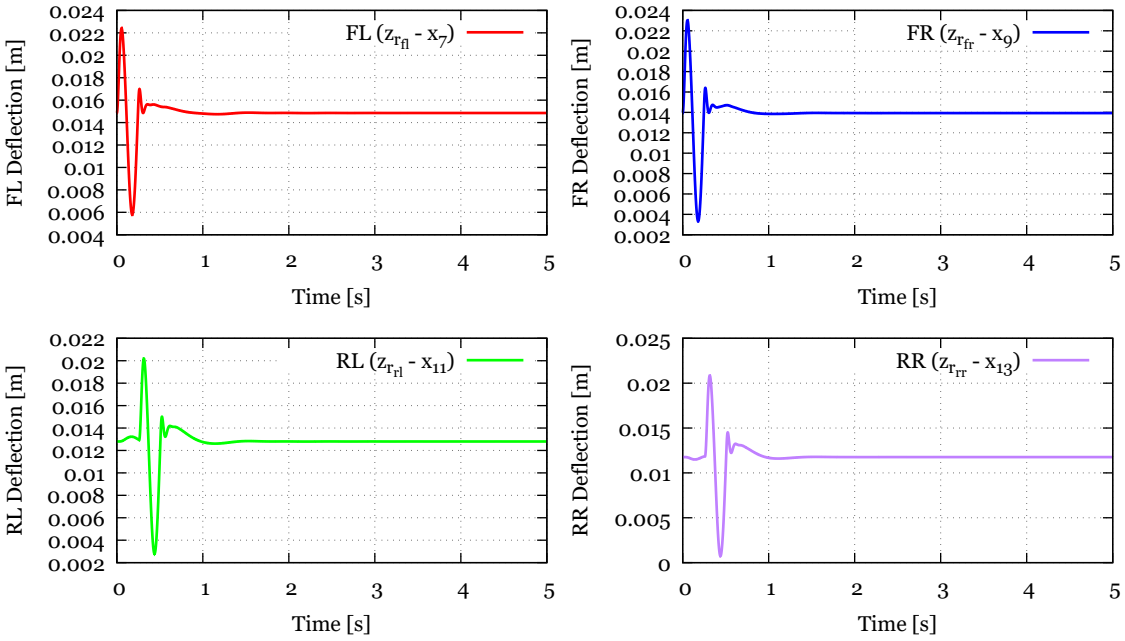


Figure 2.7: *Tire deflection $z_r - z_u$ of the four wheels during the deterministic bump. The values remain strictly positive, confirming that the tires maintain contact with the road surface and no wheel hop occurs, thus validating the use of the linear tire model.*

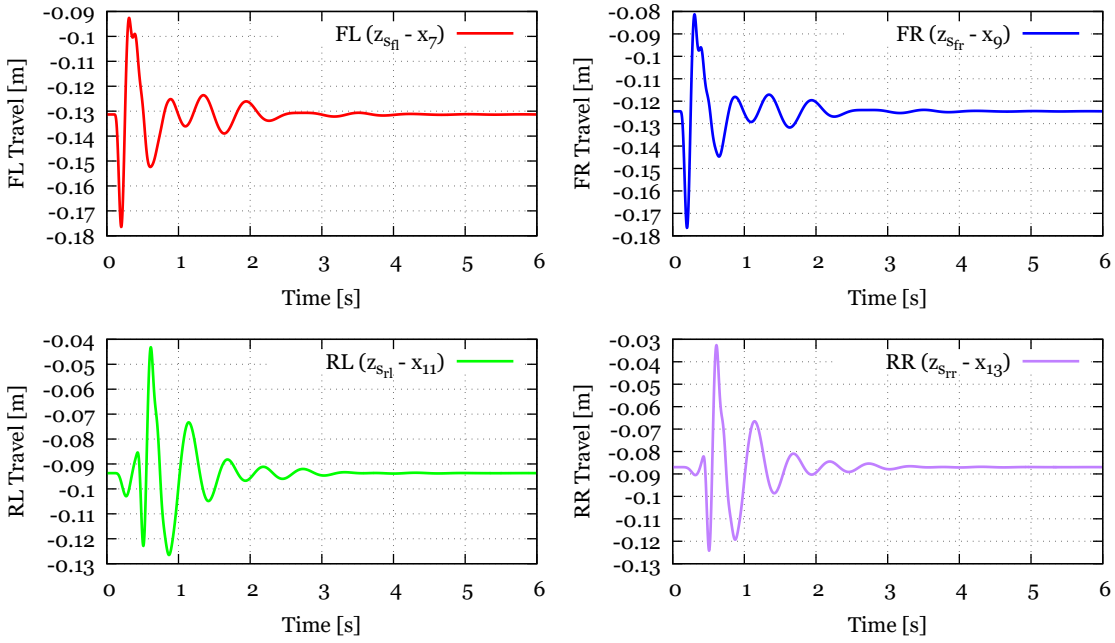


Figure 2.8: *Suspension working space (deflection) for the four corners. The calculated travel remains within typical passenger car limits ($\pm 0.08 - 0.10$ m), indicating that the vehicle would not hit mechanical stops during this maneuver.*

Chapter 3

Optimization Problem Formulation and Continuous Adjoint Method

The previous chapter presented the details of the primal solver, which computes the time evolution of the system states $\mathbf{x}(t)$ for a given set of suspension parameters. This chapter presents the formulation of the optimization problem and the derivation of the continuous adjoint system for the full-vehicle suspension model. The goal is to efficiently compute the sensitivity of a performance metric with respect to the suspension design parameters, thereby enabling gradient-based optimization of the system's dynamic behavior. The chapter begins with the definition of the design variables and objective function, followed by the mathematical derivation of the adjoint equations and their numerical solution.

3.1 Definition of the Optimization Problem

3.1.1 Ride Comfort Metrics in the Literature

Ride comfort is fundamentally characterized by the vibration transmitted to the vehicle occupants. This vibration is typically quantified by the acceleration measured at the occupant's seat or the vehicle body. Since human response to vibration is strongly frequency-dependent, measured accelerations are often "weighted" to emphasize the frequency ranges where human sensitivity is the highest [6].

The widely accepted standard for assessing whole-body vibration is ISO 2631-1:1997 [9]. This standard evaluates ride comfort using the *frequency-weighted* root-mean-square (RMS) acceleration. For vertical vibration at a seated position, the comfort index is defined as:

$$a_{wz,\mathrm{RMS}} = \sqrt{\frac{1}{t_{end} - t_{start}} \int_{t_{start}}^{t_{end}} a_{wz}(t)^2 \, \mathrm{d}t}, \tag{3.1}$$

where $a_{wz}(t) = W_k\{a_z(t)\}$ is the vertical acceleration processed through the W_k weighting filter to account for the sensitivity of the human body.

The main drawback, from a numerical optimization perspective, is that the weighting filter introduces additional dynamics that must be modeled and differentiated. When the filter is implemented in state-space form, several additional states per measurement axis are required, which increases the size and complexity of both the primal and adjoint systems. While Rimell and Mansfield [21] provide formulae to calculate the filter coefficients for digital implementation, adding these states contradicts the goal of maintaining a lean adjoint formulation. Given that the primary focus of this work is the development of the unsteady continuous adjoint methodology itself, a simpler, yet physically meaningful, ride comfort objective is required.

A common alternative is the *unweighted* squared RMS of the seat acceleration:

$$F_1 = \frac{1}{t_{end} - t_{start}} \int_{t_{start}}^{t_{end}} a_z(t)^2 \, \mathrm{d}t. \tag{3.2}$$

The appeal of this objective lies in its simplicity: the cost function is easy to compute in the time domain, smooth and compatible with gradient-based algorithms. However, unweighted RMS correlates imperfectly with subjective comfort because it treats all frequencies equally; low and high frequency vibrations can result in identical RMS values despite having different perceived comfort levels.

To address these limitations without the computational overhead of full ISO filters, this work employs a composite metric that combines acceleration with its time derivative, *jerk* ($j_z(t) = \dot{a}_z(t)$). Huang et al. [8] demonstrated that subjective discomfort correlates to both the peak and frequency of jerk. Rapid changes in acceleration (high jerk) are associated with harshness, particularly in the higher frequency range. Smoother changes are perceived as more comfortable, even if the peak acceleration is larger. A generic form of such a composite objective is:

$$F_2 = \frac{1}{t_{end} - t_{start}} \int_{t_{start}}^{t_{end}} \left(w_1 \, a_z(t)^2 + w_2 \, j_z(t)^2 \right) \, \mathrm{d}t, \tag{3.3}$$

where w_1 and w_2 are weighting coefficients. Although road inputs are random, the

frequency-selective nature of this metric can be understood by analyzing its response to a harmonic component $a_z(t) = A\sin(\omega t)$. The contribution to the cost function scales with $A^2\,(w_1 + w_2\omega^2)$. This relationship reveals an implicit frequency weighting: low-frequency components contribute mainly through the acceleration term, while higher frequencies are increasingly penalized quadratically through the jerk term. In this way, the objective function reflects the increase in discomfort associated with both large low-frequency and small high-frequency fluctuations, without the complexity of the full ISO weighting filters and the addition of extra states.

Finally, the comfort metric can be extended to account for rotational motion, since it also affects the driver's comfort. Incorporating pitch and roll accelerations into the objective function results in:

$$F_3 = \frac{1}{t_{end} - t_{start}} \int_{t_{start}}^{t_{end}} \left(w_1\, a_z(t)^2 + w_2\, j_z(t)^2 + w_3\, a_\theta(t)^2 + w_4\, a_\phi(t)^2 \right)\, \mathrm{d}t, \qquad (3.4)$$

where a_θ is the pitch acceleration and a_ϕ is the roll acceleration of the chassis.

3.1.2 Proposed Design Variables & Ride Comfort Objective Function

The optimization procedure seeks to identify the optimal suspension configuration by modifying a set of design parameters.The first step is to select the design variable vector $\mathbf{b}$, which parametrizes the physical properties of the vehicle. Modifying these parameters directly alters the system matrices $(\mathbf{A}, \mathbf{B})$, and consequently the dynamic behavior of the vehicle. In standard suspension optimization, the design variables are typically limited to the stiffness and damping coefficients of the suspension components. However, this work also includes certain inertial characteristics of the chassis as design variables. By allowing parameters such as the distance of the front axle to the CoG of the vehicle (a) or the pitch moment of inertia (I_{yy}) to vary, the optimization process evolves from a tuning exercise into a preliminary conceptual design tool. This choice allows the optimization to transform the vehicle's layout to a design that best suits objective goals to the specific road environment. The vector of design variables is therefore defined as:

$$\mathbf{b} = \begin{bmatrix} K_{s_f} & K_{s_r} & B_{s_f} & B_{s_r} & I_{yy} & a \end{bmatrix}^T \qquad (3.5)$$

After converging the primal equations at each time-step, the instantaneous value of the objective function $f(\mathbf{x}(\mathbf{b}), \mathbf{b}, t)$ is computed. For this unsteady problem, the objective function F is defined as the time integral of f over a time interval starting from time-step N_{start} (time t_{start}) and ending at the final time-step N_{end} (time t_{end}).

For uniform time-steps Δt, the objective function is approximated numerically as

$$F(\mathbf{x}(\mathbf{b}), \mathbf{b}) = \int_{t_{start}}^{t_{end}} f(\mathbf{x}(\mathbf{b}), \mathbf{b}, t)\, dt \tag{3.6}$$

The second step is to define the objective function that fits the purpose of this work. The goal in this case is for the vehicle suspension to ensure ride comfort, that is, minimize the vertical vibrations transmitted to the driver's seat. A compromise is adopted between the physiologically detailed ISO 2631 measure and the unweighted RMS of acceleration and jerk signals. The objective function is formulated as a weighted sum of the driver's seat vertical acceleration, vertical jerk, and the chassis pitch and roll accelerations and is defined as:

$$F = \frac{1}{2\, t_{period}} \int_{t_{start}}^{t_{end}} \left(w_1 \dot{x}_{16}^2 + w_2 \ddot{x}_{16}^2 + w_3 \dot{x}_4^2 + w_4 \dot{x}_6^2 \right) dt, \tag{3.7}$$

where $w_1, \ldots, w_4$ are weighting coefficients and $t_{period} = t_{end} - t_{start}$ is the duration of the integral. To facilitate the adjoint formulation, the kinematic quantities are expressed as algebraic functions of the state vector $\mathbf{x}$, the input vector $\mathbf{u}$, and the system matrices.

Driver's Seat Vertical Acceleration The DS vertical velocity of the driver's seat corresponds to the state derivative $\dot{x}_{16}$. Using the state-space equation $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} + S$, this term is isolated using the 16th row of the system matrix:

$$\dot{x}_{16} = \mathbf{A_{16}}\mathbf{x} + S_{16}, \tag{3.8}$$

where $A_{16} = [A_{16,1}, A_{16,2}, \ldots, A_{16,16}]$ is the 16th row vector of matrix A, and $S_{16} = -g$ is the 16th element of the source vector S. Note that the 16th row vector of the input matrix is zero ($B_{16} = \mathbf{0}$).

Driver's Seat Vertical Jerk The DS vertical jerk $\ddot{x}_{16}$ is the time derivative of the acceleration. Differentiating the state equation yields:

$$\ddot{x}_{16} = \frac{d}{dt}(A_{16}\mathbf{x} + S_{16}) = A_{16}\dot{\mathbf{x}}. \tag{3.9}$$

Substituting the full state equation back into this expression results in:

$$\ddot{x}_{16} = A_{16}(A\mathbf{x} + B\mathbf{u} + S) = \mathbf{J_1}\mathbf{x} + \mathbf{J_2}\mathbf{u} + J_3, \tag{3.10}$$

Chassis Pitch and Roll Angular Acceleration The pitch acceleration ($\ddot{\theta}$) and roll acceleration ($\ddot{\phi}$) correspond to the time derivatives of the pitch velocity ($\dot{x}_4$) and roll velocity ($\dot{x}_6$), respectively. These are expressed using the 4th and 6th rows of the system matrix:

$$\dot{x}_4 = \mathbf{A}_4\mathbf{x}, \quad \dot{x}_6 = \mathbf{A}_6\mathbf{x}, \tag{3.11}$$

where A_4 and A_6 are the 4th and 6th row vectors of matrix A. In both accelerations the corresponding row vectors of B and S are zero.

Substituting these vector expressions into the objective function yields the final form used for optimization:

$$F = \frac{1}{2\,t_{period}} \int_{t_{start}}^{t_{end}} \left[w_1(\mathbf{A_{16}}\mathbf{x} + S_{16})^2 + w_2(J_1\mathbf{x} + J_2\mathbf{u} + J_3)^2 + w_3(A_4\mathbf{x})^2 + w_4(A_6\mathbf{x})^2 \right] dt. \tag{3.12}$$

The weighting coefficients w_1 through w_4 are selected to normalize the metrics due to their different units and orders of magnitude. While these parameters can be adjusted to prioritize specific metrics, in this work, they are determined so that all comfort metrics are considered equally important.

3.2 Formulation of the Continuous Adjoint Method

To employ gradient-based optimization methods, the gradients of the objective function w.r.t. the design variables must be computed. The primal system of equations is written in a residual form as:

$$\mathbf{R} = \mathbf{R}(\mathbf{x}(\mathbf{b}), \mathbf{b}) = \dot{\mathbf{x}} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u} - \mathbf{S} = 0 \tag{3.13}$$

The mathematical formulation of the continuous adjoint method starts by defining the augmented objective function (or Lagrangian) using the full weighted metric F derived in Eq. 3.12:

$$F_{aug} = F + \int_{t_{start}}^{t_{end}} \mathbf{\Psi}^T \mathbf{R} dt = \frac{1}{2\,t_{period}} \int_{t_{start}}^{t_{end}} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{b}) dt + \int_{t_{start}}^{t_{end}} \mathbf{\Psi}^T \left(\dot{\mathbf{x}} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u} - \mathbf{S} \right) dt \tag{3.14}$$

where $\mathcal{L}$ represents the weighted sum of the squared comfort metrics and $\mathbf{\Psi} \in \mathbb{R}^{16\times 1}$ is the vector of adjoint variables. The first term represents the performance measure, while the second term enforces the constraints of the system (i.e. the governing equations). The sensitivity of the augmented objective function w.r.t the design variables $\mathbf{b}$ is obtained by differentiating Eq. 3.14. Integrating the term $\mathbf{\Psi}^T \cdot (\delta\dot{\mathbf{x}}/\delta\mathbf{b})$ by parts transfers the time derivative to the adjoint variables, resulting in a term involving $-\dot{\mathbf{\Psi}}$ and boundary terms. Substituting this result and grouping the terms

multiplying $\delta\mathbf{x}/\delta\mathbf{b}$ yields the **Field Adjoint Equations** (FAE). To eliminate the computationally expensive state sensitivities, the coefficient of $\delta\mathbf{x}/\delta\mathbf{b}$ need to vanish. This condition defines the adjoint system:

$$\frac{d\mathbf{\Psi}^T}{dt} = -\mathbf{\Psi}^T\mathbf{A} + \frac{1}{2t_{period}}\mathbf{Q}(t) \tag{3.15}$$

The forcing term $\mathbf{Q}(t)$ is the vector that is assembled by the four components of the objective function:

1. Driver's Seat Acceleration: From Eq. 3.8, the contribution is:

$$\mathbf{Q_{seat-acc}} = 2w_1(\mathbf{A_{16}x} + S_{16})\mathbf{A_{16}} \tag{3.16}$$

2. Driver's Seat Jerk: From Eq. 3.10, the contribution is:

$$\mathbf{Q_{seat-jerk}} = 2w_2[\mathbf{A_{16}}(\mathbf{Ax} + \mathbf{Bu} + \mathbf{S})]\mathbf{A_{16}A} \tag{3.17}$$

3. Pitch and Roll Acceleration: From Eq. 3.11, the contribution is:

$$\mathbf{Q_{pitch-acc}} = 2w_3(\mathbf{A_4x})\mathbf{A_4} \tag{3.18}$$
$$\mathbf{Q_{roll-acc}} = 2w_4(\mathbf{A_6x})\mathbf{A_6} \tag{3.19}$$

The adjoint system requires a terminal boundary condition to eliminate the remaining boundary terms from the integration by parts. The term that contains $\delta\mathbf{x}/\delta\mathbf{b}$ at $t = t_{end}$ can not be known a priori. This defines the **Adjoint Boundary Condition** (ABC):

$$\mathbf{\Psi}^T\big|_{t_{end}} = \mathbf{0} \tag{3.20}$$

The adjoint system, (Eqs. 3.15 and 3.20), requires backward integration over the time interval $[t_{start}, t_{end}]$. Typically to compute the adjoint states at t_{k-1}, solvers require access to the primal states at the same time t_{k-1} and the adjoint states at the previosu time t_k.

$$\mathbf{\Psi}(t_{k-1}) = function(\mathbf{\Psi}(t_k), \mathbf{x}(t_{k-1})) \tag{3.21}$$

The standard Runge-Kutta 4 (RK4) integration scheme used to solve the FAE relies on evaluating the system dynamics at intermediate time steps (specifically at $t - \frac{\Delta t}{2}$) to compute the internal slopes. However, a numerical challenge arises during the backward pass (stepping from t_k to t_{k-1}): the forcing term $Q(x(t))$, which depends

on the primal state, is only defined at the discrete time points t_k and t_{k-1}. A Zero-Order Hold approximation (assuming $Q(x(t))$ remains constant over the interval $[t_k,t_{k-1}]$) was rejected, as it effectively reduces the integration accuracy of the term subsequently to the gradient calculation. To resolve this a *First-Order Hold* (Linear Interpolation) strategy is applied. For any local time t_{local} within the backward integration step $[t_{k-1}, t_k]$, the forcing term is approximated as:

$$Q(t_{local}) \approx (1-\alpha)Q(x(t_k)) + \alpha Q(x(t_{k-1})) \tag{3.22}$$

where $\alpha = (t_k - t_{local})/\Delta t$. This approach uses the first order information from the completed forward step (x_{k-1} and x_k) to reduce the approximation error. To validate this strategy, the gradients have been also calculated through Finite Differences method. The results that are shown in the Table 3.1 prove the strategy's effectiveness and accuracy.

Table 3.1: *Verification of Adjoint Gradients vs. Finite Differences.*

Var	**Adjoint**	**Finite Diff**	**Diff**	**Error %**
K_{s_f}	1.04945×10^{-5}	1.04948×10^{-5}	-3.30×10^{-10}	0.0031
K_{s_r}	2.16640×10^{-7}	2.16679×10^{-7}	-3.87×10^{-11}	0.0179
B_{s_f}	-4.80526×10^{-5}	-4.80524×10^{-5}	-2.06×10^{-10}	0.0004
B_{s_r}	-3.19176×10^{-5}	-3.19179×10^{-5}	3.54×10^{-10}	0.0011
I_{yy}	-7.19632×10^{-5}	-7.19649×10^{-5}	1.75×10^{-9}	0.0024
a	2.18728×10^{-1}	2.18736×10^{-1}	-7.34×10^{-6}	0.0034

Once the adjoint system is solved backwards, the **Sensitivity Derivatives** (SD) of the objective function with respect to the design variables $\mathbf{b}$ are computed as:

$$\frac{\delta F}{\delta \mathbf{b}} = \int_{t_{start}}^{t_{end}} \left[\mathbf{\Psi}^T \left(-\frac{\delta A}{\delta \mathbf{b}} \mathbf{x} \right) + \frac{1}{2t_{period}} \mathbf{G}(t) \right] dt - \mathbf{\Psi}^T \frac{\delta \mathbf{x}}{\delta \mathbf{b}} \bigg|_{t_{start}} \tag{3.23}$$

where $\mathbf{G}(t)$ is the gradient vector that contains terms that derive from the differentiation of the four component metrics of the objective function:

1. Driver's Seat Acceleration:

$$\mathbf{G_{seat-acc}} = 2w_1(\mathbf{A_{16}x} + S_{16}) \frac{\delta \mathbf{A_{16}}}{\delta \mathbf{b}} \mathbf{x} \tag{3.24}$$

2. Driver's Seat Jerk:

$$\mathbf{G_{seat-jerk}} = 2w_2 \left[\mathbf{A_{16}}(\mathbf{Ax} + \mathbf{Bu} + \mathbf{S}) \right] \left[\frac{\delta \mathbf{A_{16}}}{\delta \mathbf{b}} \dot{\mathbf{x}} + \mathbf{A_{16}} \frac{\delta \mathbf{A}}{\delta \mathbf{b}} \mathbf{x} \right] \tag{3.25}$$

3. Pitch and Roll Acceleration:

$$\mathbf{G_{pitch-acc}} = 2w_3(\mathbf{A_4x})\frac{\delta \mathbf{A_4}}{\delta \mathbf{b}}\mathbf{x} \tag{3.26}$$

$$\mathbf{G_{roll-acc}} = 2w_4(\mathbf{A_6x})\frac{\delta \mathbf{A_6}}{\delta \mathbf{b}}\mathbf{x} \tag{3.27}$$

The problem lies in the fact that the term $\frac{\delta \mathbf{x}}{\delta \mathbf{b}}\big|_{t_{start}}$ is not available at the current stage. However, since all simulations begin at $t_{start} = 0$, the vehicle is assumed to be at a steady-state equilibrium. Consequently, the initial state vector $\mathbf{x}(t_{start}) = \mathbf{x}(0) = \mathbf{x}_{ss}$ satisfies the algebraic equation:

$$\mathbf{A}\mathbf{x}(t_{start}) + \mathbf{S} = \mathbf{0} \tag{3.28}$$

By differentiating this equilibrium condition with respect to the design variables vector $\mathbf{b}$, we can directly compute the required initial sensitivity term:

$$\mathbf{A}\left.\frac{\delta \mathbf{x}}{\delta \mathbf{b}}\right|_{t_{start}} + \frac{\delta \mathbf{A}}{\delta \mathbf{b}}\,\mathbf{x}|_{t_{start}} = -\frac{\delta \mathbf{S}}{\delta \mathbf{b}} \tag{3.29}$$

The right-hand side is zero, so the equation yields to:

$$\mathbf{A}\left.\frac{\delta \mathbf{x}}{\delta \mathbf{b}}\right|_{t_{start}} = -\frac{\delta \mathbf{A}}{\delta \mathbf{b}}\,\mathbf{x}|_{t_{start}} \tag{3.30}$$

The above linear system 3.30 is solved using the Gaussian elimination method with partial pivoting from Algorithm 1.

The complete sequence for computing the sensitivity derivatives is summarized in Algorithm 3.

Once $\delta F/\delta \mathbf{b}$ is computed, it can be used together with any gradient based method to update the design variables.

Algorithm 3 Computation of Sensitivity Derivatives via Continuous Adjoint

Require: Primal states solution $\mathbf{x}(t)$, System Matrices $(\mathbf{A}, \mathbf{B}, \mathbf{S})$, Design variables vector $\mathbf{b}$

Ensure: Sensitivity derivatives $\nabla_{\mathbf{b}} F$

Step 1: Initial Sensitivity (Steady-State)

Compute RHS: $\mathbf{B}_{\text{rhs}} \leftarrow -\frac{\delta \mathbf{A}}{\delta \mathbf{b}} \mathbf{x}(t_{\text{start}})$

Solve for initial sensitivity $\frac{\delta \mathbf{x}}{\delta \mathbf{b}}\big|_{t_{start}}$ using Gauss Elimination: $\mathbf{A} \cdot \frac{\delta \mathbf{x}}{\delta \mathbf{b}}\big|_{t_{start}} = \mathbf{B}_{\text{rhs}}$

Step 2: Adjoint Analysis (Backward Integration)

Initialize Adjoint (Terminal Condition): $\mathbf{\Psi}^T(t_{\text{end}}) \leftarrow \mathbf{0}$

Initialize Gradient Accumulator: $\mathbf{G} \leftarrow \mathbf{0}$

Pre-compute Adjoint Forcing $\mathbf{Q}_{curr}$ at t_{end}

$N_{\text{steps}} \leftarrow (t_{\text{end}} - t_{\text{start}})/\Delta t$

for $k = N_{\text{steps}}$ down to 1 **do**

  $t_k \leftarrow t_{\text{start}} + k \cdot \Delta t$

  Retrieve primal state $\mathbf{x}_{\text{prev}} \leftarrow \mathbf{x}(t_{k-1})$

  Compute Adjoint Forcing $\mathbf{Q}_{next}$ at t_{k-1}

  Accumulate Gradient Contribution:

  Determine trapezoidal weight: $w \leftarrow \begin{cases} 0.5 & \text{if } k = N_{\text{steps}} \\ 1.0 & \text{otherwise} \end{cases}$

  Calculate integrand I_k using $\mathbf{x}(t_k)$ and $\mathbf{\Psi}(t_k)$

  $\mathbf{G} \leftarrow \mathbf{G} + I_k \cdot (w \cdot \Delta t)$

  Backward Step (RK4):

  Integrate $\frac{d\mathbf{\Psi}}{dt}^T = -\mathbf{\Psi}^T \mathbf{A} + Q$ from t_k to t_{k-1}

  $\mathbf{\Psi}(t_{k-1}) \leftarrow \text{RK4\_Step}(\mathbf{\Psi}(t_k), \mathbf{Q}_{\text{curr}}, \mathbf{Q}_{\text{prev}}, -\Delta t)$

  Update Forcing: $\mathbf{Q}_{\text{curr}} \leftarrow \mathbf{Q}_{\text{prev}}$

end for

Step 3: Final Assembly

Accumulate Initial Time Contribution:

Calculate integrand term I_0 using $\mathbf{x}(t_{\text{start}})$ and $\mathbf{\Psi}(t_{\text{start}})$

$\mathbf{G} \leftarrow \mathbf{G} + I_0 \cdot (0.5 \cdot \Delta t)$ ▷ Trapezoidal boundary

Apply Boundary Correction:

$\nabla_{\mathbf{b}} F = \mathbf{G} - \mathbf{\Psi}^T(t_{\text{start}}) \cdot \frac{\partial \mathbf{x}}{\partial \mathbf{b}}|_{t_{start}}$

return $\nabla_{\mathbf{b}} F$

3.3 Gradient-based optimization method

The sensitivity analysis described in the previous section provides the gradients of the objective function ∇F with respect to the design variables. To identify the optimal suspension configuration, this gradient information is utilized within an iterative optimization framework. The fundamental update rule for the design variables $\mathbf{b}$ at iteration k is defined as:

$$\mathbf{b}^{(k+1)} = \mathcal{P}_\Omega \left(\mathbf{b}^{(k)} + \alpha^{(k)} \, \mathbf{p}^{(k)} \right) \tag{3.31}$$

where $\mathbf{p}^k$ represents the search direction, α^k is the scalar step length and $\mathcal{P}_\Omega$ is the projection operator that controls the bounds of the variables. Since the design variables in this study are subject to physical limitations defined by lower (b^L) and upper (b^U) bounds, the problem is classified as a Bound-Constrained Optimization problem.

3.3.1 Handling Constraints: The Projection Method

Two primary methodologies exist in the literature for handling constraints: methods that convert the constrained problem into an unconstrained one via penalty terms (e.g., Augmented Lagrangian Methods), and methods that explicitly enforce bounds during the search. For problems governed by simple box constraints ($\mathbf{b}_i^L \leq \mathbf{b}_i \leq \mathbf{b}_i^U$), complex penalty methods often introduce unnecessary computational overhead. According to Nocedal and Wright [19], the **Projected Gradient** strategy is the most efficient approach for such constraints. Consequently, this work employs a Projection Operator $\mathcal{P}_\Omega$ that maps any vector back into the feasible design domain Ω:

$$\mathcal{P}_\Omega(\mathbf{b}_i) = \begin{cases} \mathbf{b}_i^L & \text{if } \mathbf{b}_i < \mathbf{b}_i^L \\ \mathbf{b}_i & \text{if } \mathbf{b}_i^L \leq \mathbf{b}_i \leq \mathbf{b}_i^U \\ \mathbf{b}_i^U & \text{if } \mathbf{b}_i > \mathbf{b}_i^U \end{cases} \tag{3.32}$$

This projection is applied immediately during the line search phase, ensuring that the primal solver (Runge-Kutta 4) is never executed with physically invalid parameters.

3.3.2 Search Direction: Inverse BFGS with Scaling

The selection of the search direction $\mathbf{p}^{(k)}$ is critical for convergence speed. The simplest approach, the Steepest Descent method ($\mathbf{p}^{(k)} = -\nabla F^{(k)}$), is computationally

inexpensive but is known to exhibit slow convergence rates due to "zig-zagging" in problems with ill-conditioned curvature. Conversely, Newton's method utilizes the exact Hessian matrix ($\nabla^2 F$) to achieve quadratic convergence but is computationally prohibitive for this application, as calculating the exact second derivatives of the unsteady adjoint system is excessively expensive.

To balance computational efficiency with convergence speed, this work employs a Quasi-Newton approach. Specifically, the **BFGS (Broyden-Fletcher-Goldfarb-Shanno)** algorithm is used to approximate the inverse Hessian matrix $\mathbf{H} \approx (\nabla^2 F)^{-1}$. The approximate Hessian $\mathbf{B}^{(k)}$ satisfies the secant equation:

$$\mathbf{B}^{(k+1)}\mathbf{s}^{(k)} = \mathbf{y}^{(k)} \tag{3.33}$$

where the difference vectors are defined as:

$$\begin{aligned} \mathbf{s}^{(k)} &= \mathbf{b}^{(k+1)} - \mathbf{b}^{(k)} \\ \mathbf{y}^{(k)} &= \nabla F(\mathbf{b}^{(k+1)}) - \nabla F(\mathbf{b}^{(k)}) \end{aligned} \tag{3.34}$$

In order to avoid the computational cost of solving linear systems at every iteration, the matrix $\mathbf{H}$ is updated using the Inverse BFGS formula:

$$\mathbf{H}^{(k+1)} = \left[\mathbf{I} - \rho^{(k)}\mathbf{s}^{(k)}(\mathbf{y}^{(k)})^T\right]\mathbf{H}^{(k)}\left[\mathbf{I} - \rho^{(k)}\mathbf{y}^{(k)}(\mathbf{s}^{(k)})^T\right] + \rho^{(k)}\mathbf{s}^{(k)}(\mathbf{s}^{(k)})^T \tag{3.35}$$

where $\rho^{(k)} = \frac{1}{(\mathbf{y}^{(k)})^T\mathbf{s}^{(k)}}$. Finally, the search direction at iteration k is computed as:

$$\mathbf{p}^{(k)} = -\mathbf{H}^{(k)}\nabla F^{(k)} \tag{3.36}$$

Variable Scaling

A significant numerical challenge in optimizing full-vehicle suspension systems is the vast disparity in variable magnitudes. Suspension stiffness coefficients (K_{s_f}, K_{s_r}) are typically in the order of 10^4 N/m, while the variable of the front axle to CoG distance a is in the order of 10^0 m. If unscaled, the optimization would be dominated by the stiffness gradients, effectively stalling the evolution of a.

To address this, the initial inverse Hessian $\mathbf{H}^{(0)}$ is not set to the identity matrix, but to a **diagonal scaling matrix** containing specific weighting factors η:

$$H^{(0)} = \mathrm{diag}(\eta_1, \eta_2, \ldots, \eta_N) \tag{3.37}$$

These scaling factors normalize the expected step sizes, ensuring that variables with

smaller magnitudes are not numerically overshadowed. In the implementation, this mechanism is also used to "freeze" specific variables (e.g., I_{yy}) by setting their corresponding scaling factor to zero ($\eta_i = 0$), thereby forcing the search direction for that variable to zero. This is particularly useful when the optimization is focused on a certain number of design variables.

Curvature Safeguard

To maintain numerical stability, a curvature safeguard is implemented. The update of the approximate inverse Hessian matrix is performed only if the curvature condition $(\mathbf{s}^{(k)})^T \mathbf{y}^{(k)} > \epsilon$ is satisfied (typically $\epsilon = 10^{-10}$). If this condition fails, it indicates that the step did not capture sufficient positive curvature. In such cases, the BFGS update is skipped ($\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)}$) to preserve the positive definiteness of the matrix.

3.3.3 Step Length and Termination Criteria

Once the direction is established, $\alpha^{(k)}$ is determined. While exact line searches are computationally expensive, inexact line searches that satisfy the **Armijo condition** (Sufficient Decrease) are standard practice. The Armijo condition requires that the reduction in F be proportional to the step length and the directional derivative:

$$F(\mathbf{b}_{trial}) \leq F(\mathbf{b}^{(k)}) + c_1 \, \alpha \left(\nabla F^{(k)}\right)^T \mathbf{p}^{(k)} \tag{3.38}$$

where $c_1 \in (0, 1)$ is a small constant (typically 10^{-4}). To find an α that satisfies this, a **Backtracking** strategy is employed. The algorithm attempts a full step ($\alpha = 1.0$). If the Armijo condition is not met, α is reduced by a contraction factor $\rho \in (0, 1)$ until the condition holds. Crucially, the trial point is projected onto the bounds *before* the objective function is evaluated:

$$\mathbf{b}_{trial} = \mathcal{P}_\Omega(\mathbf{b}^{(k)} + \alpha \mathbf{p}^{(k)}) \tag{3.39}$$

This ensures that the objective function is only evaluated within the feasible bounds.

Termination Criteria: The optimization loop continues until convergence is detected or a maximum number of cycles is reached. Convergence is defined based on the relative stability of the objective function cost. The process terminates if the relative change in cost between consecutive iterations falls below a specified tolerance ϵ_F:

$$\frac{|F^{(k+1)} - F^{(k)}|}{|F^{(k)}|} < \epsilon_F \tag{3.40}$$

In the numerical experiments presented, the tolerance is set to $\epsilon_F = 10^{-5}$. The complete optimization procedure is summarized in Algorithm 4.

Algorithm 4 Projected BFGS Optimization

Require: Initial design $\mathbf{b}^{(0)}$, Bounds $\mathbf{b}^L, \mathbf{b}^U$, Scaling factors η, Tolerance ϵ_F, Curvature tolerance ϵ, Max Cycles K_{max}, ρ
Ensure: Optimal design $\mathbf{b}^*$

Initialize:
$\mathbf{b} \leftarrow \mathbf{b}^{(0)}$
$\mathbf{H} \leftarrow \text{diag}(\eta)$
$k \leftarrow 0$
while $k < K_{max}$ **do**
1. Gradient Evaluation:
Compute objective function $F^{(k)}$ and gradient $\nabla F^{(k)}$ using Adjoint Method 3
2. Hessian Correction:
if $k > 0$ **then**
$\mathbf{s} \leftarrow \mathbf{b}^{(k)} - \mathbf{b}^{(k-1)}$
$\mathbf{y} \leftarrow \nabla F^{(k)} - \nabla F^{(k-1)}$
if $\mathbf{s}^T\mathbf{y} > \epsilon$ **then** ▷ Curvature Safeguard
Update $\mathbf{H}$ using Eq. (3.35)
end if
end if
3. Search Direction:
$\mathbf{p}^{(k)} \leftarrow -\mathbf{H}\nabla F^{(k)}$
4. Projected Line Search (Armijo):
$\alpha \leftarrow 1.0$
repeat
$\mathbf{b}_{trial} \leftarrow \mathcal{P}_\Omega(\mathbf{b}^{(k)} + \alpha\mathbf{p}^{(k)})$
$F_{trial} \leftarrow F(\mathbf{b}_{trial})$
if $F_{trial} \leq F^{(k)} + c_1\alpha(\mathbf{p}^{(k)})^T\nabla F^{(k)}$ **then**
break
end if
$\alpha \leftarrow \alpha \cdot \rho$
until $\alpha < 10^{-6}$
5. Termination Check:
if $k > 0$ **and** $|F_{trial} - F^{(k)}|/|F^{(k)}| < \epsilon_F$ **then**
Converged
break
end if
6. Update Step:
$\mathbf{b}^{(k+1)} \leftarrow \mathbf{b}_{trial}$
$k \leftarrow k + 1$
end while
return $\mathbf{b}^{(k)}$

Chapter 4

Management of Storage Requirements: Check-pointing

As outlined in the Introduction, the primary computational challenge in the unsteady adjoint method is the storage requirement for the primal state history. While the vehicle model developed in Chapter 2 is relatively "lightweight", the objective of this framework is to support high-fidelity models in which storing the full time history is impractical due to memory constraints. This chapter details the specific memory management strategy implemented to solve this limitation: a Check-pointing algorithm proposed in [17].

4.1 Storage Requirements in Unsteady Adjoint Methods

As derived in Chapter 3, the Field Adjoint Equations (FAE) (Eq. 3.15) require the evaluation of the forcing term $\mathbf{Q}(t)$ during the backward integration process (from t_{end} to t_{start}). This forcing term is state-dependent, meaning that to compute $d\Psi/dt$ at any time instance t, the corresponding primal state vector $\mathbf{x}(t)$ must be available:

$$\frac{d\boldsymbol{\Psi}^T}{dt} = -\boldsymbol{\Psi}^T\mathbf{A} + \frac{1}{2t_{\text{period}}}\mathbf{Q}(\mathbf{x}(t)) \tag{4.1}$$

For a simulation with N_{steps} time steps, the most straightforward approach is the *Full Storage* strategy, where the entire history of state vectors $\left[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N_{steps}}\right]$

is saved to memory during the primal solution. For the simplified 8-DoF vehicle model, this is feasible.

In the context of high-fidelity engineering problems such as Computational Fluid Dynamics (CFD), while the number of physical variables is small (e.g., 4 variables for incompressible flow: pressure and three velocity components), they would have to be stored for every cell or node in the computational mesh. Consequently, the dimension N of the global state vector is the product of the number of physical variables and the mesh size ($N = N_{vars} \times N_{cells}$). Mesh sizes can reach up 10^8 cells, resulting in a state vector of size $N = 4 \times 10^8$ per single time step. So the total memory requirement scales as $M \propto (N_{vars} \cdot N_{cells}) \cdot N_{steps}$.

The opposite extreme is the *Zero Storage* strategy. Here, only the initial condition $\mathbf{x}_0$ is stored. Whenever the adjoint solver requires a state $\mathbf{x}_k$, the primal solver re-integrates from $t = 0$ to $t = t_k$. Although this minimizes memory usage to $\mathcal{O}(1)$, the computational effort scales quadratically, $Cost \propto O(N_{steps}^2)$, rendering it prohibitive for long-duration simulations.

To resolve this trade-off, this work implements a *Check-pointing* strategy. This method stores a limited set of snapshot solutions (checkpoints) in a fixed-size s storage. During the backward adjoint integration, missing states are recovered by re-integrating the primal equations starting from the nearest stored checkpoint.

4.2 The Check-pointing Algorithm

While classical strategies exist, such as Binomial Check-pointing [5], they typically require the total number of time steps N_{steps} to be fixed and known a priori. To maintain the generality of the framework—allowing for adaptive time stepping solvers in the future—this work implements the Check-pointing Algorithm proposed in [17]. The core concept of the algorithm is the assignment of a **Level** (l) to each saved state. A checkpoint is defined as a triplet stored in memory:

$$cp_i = \{i, l_i, \mathbf{x}(i)\} \tag{4.2}$$

where:

- $i \in [0, N_{steps}]$ is the time step index of the recorded state.
- $l \in \mathbb{N}$ is the level of the checkpoint.
- $\mathbf{x} \in \mathbb{R}^{16}$ is the saved state vector.

The set of all stored checkpoints forms the buffer C. The algorithm enforces a maximum buffer capacity of S.

$$\mathcal{C} = [cp_0, cp_1, \ldots, cp_{S-1}] \tag{4.3}$$

4.2.1 The Forward Sweep (Recording Strategy)

As the primal system is integrated forward in time, the algorithm dynamically decides which states to save based on the concept of a **dispensable checkpoint**. A checkpoint $cp_i \in C$ is defined as dispensable if there exists another checkpoint $cp_j \in C$ stored at a later time ($t_j > t_i$) that possesses a strictly higher level ($l_j > l_i$):

$$\text{is\_dispensable}(cp_i) \iff \exists cp_j \in C : (t_j > t_i) \wedge (l_j > l_i)$$

At every time step t_k, a new candidate checkpoint $\{k, 0, \mathbf{x}(k)\}$ is created. The buffer update follows three distinct cases as the forward simulation progresses:

1. **Filling**: If the buffer is not full ($|C| < S$), the new candidate is simply added to the buffer with level $l = 0$. Note that the initial condition ($t = 0$) is assigned an infinite level ($l = \infty$) to ensure it is never treated as dispensable and remains in the buffer throughout the simulation $cp_0 = \{0, \infty, \mathbf{x}(t = 0)\}$.
2. **Replacement**: If the storage is full and at least one dispensable checkpoint exists, the buffer is scanned from right to left (latest to earliest) and the **rightmost** (latest) dispensable checkpoint is removed.
3. **Promotion**: If the storage is full and no checkpoint is dispensable, the algorithm must extend the final time step. The final checkpoint in the buffer is overwritten by the new state, and its level is incremented: $l_{new} = l_{old} + 1$.

4.2.2 The Backward Sweep (Adjoint Retrieval)

During the adjoint (backward) integration, the solver requires the primal state vector $\mathbf{x}(t)$ in reverse chronological order ($t_{N_{steps}}, t_{N_{steps}-1}, \ldots, t_0$). When the adjoint solver requests a state at time t_{req} the algorithm checks the checkpoint buffer. If the state is available, it is retrieved directly. However, if the required state is missing, the algorithm must reconstruct it using the available checkpoints. The retrieval logic is the following:

1. **Release**: After the adjoint step at time t_{req+1} is completed, the primal state $\mathbf{x}(t_{req+1})$ is typically no longer needed. If this state corresponds to the checkpoint currently at the top of the buffer (the latest stored time), it is removed to free space.

2. **Locate**: The algorithm retrieves the nearest stored checkpoint $c_{near.cp}$ such that $t_{near.cp} < t_{req}$.

3. **Recompute & Refill**: To bridge the gap between $t_{near.cp}$ and t_{req}, the primal solver is restarted from $\mathbf{x}_{near.cp}$ and integrated forward. During this re-integration, the generated intermediate states are passed back into the Forward Sweep logic (Section 4.2.1). Since the buffer typically has free space (from the *Release* step), these intermediate states are captured as new $l = 0$ checkpoints. This fills in the gap with snapshots, ensuring that subsequent requests for $t_{req} - 1, t_{req} - 2, \ldots$ can be met directly from the buffer without immediate re-computation.

4.2.3 Numerical Implementation of the Algorithm

A specific challenge arises in the numerical implementation regarding the Adjoint Forcing term. The First-Order formulation (Eq. 3.22) requires the simultaneous availability of two primal states, $\mathbf{x}(t_k)$ and $\mathbf{x}(t_{k-1})$, to accurately interpolate $\mathbf{Q}(t)$ across the integration interval $[t_k, t_{k-1}]$. However, the standard Check-pointing algorithm typically retrieves a single snapshot at a time. To resolve this, the adjoint solver utilizes a strategy. It maintains $\mathbf{x}_{current}$ (the state at t_k, preserved from the previous iteration) and requests $\mathbf{x}_{target}$ (the state at t_{k-1}) from the checkpoint buffer. The complete procedure is detailed in Algorithm 5. The function `advance` is used to handle the buffer updates according to the Forward Sweep 4.2.1. Also, the detailed math for the gradient accumulation $\mathbf{G}$ and the objective function F is avoided here to focus on the check-pointing logic.

To visualize the flow of the algorithm, we consider a simplified case with $N_{steps} = 10$ time steps and a buffer capacity of $S = 5$. Figure 4.1 shows the state of the buffer and the solver actions throughout the process. During the forward pass (blue line), the algorithm initially fills the buffer with $l = 0$ checkpoints (except the initial one). When the buffer reaches capacity, the algorithm promotes the next state $idx = 5$ to a higher level $l = 1$. This high-level state remains in memory, while older $l = 0$ checkpoints are released so that newer ones with with the same $l = 0$ can take their place. The buffer fills again for the second time, and the state $idx = 9$ gets promoted to $l = 1$ while intermediate $l = 0$ checkpoints are released. Once the forward simulation ends, the buffer looks like this:

$$\mathcal{C} = [\{0, \infty\}, \{5, 1\}, \{6, 0\}, \{9, 1\}, \{10, 0\}] \tag{4.4}$$

The backward adjoint integration starts (red line) in reverse chronological order. For the final steps ($t = 10$ and $t = 9$), the primal states are retrieved directly from the buffer and then released. However, as the solver steps back to $t = 8$, it encounters a gap where the states were released to save memory. To resolve this, the algorithm identifies the nearest checkpoint— $l = 0$ checkpoint at $idx = 6$. It then does a

Algorithm 5 Computation of Sensitivity Derivatives via Continuous Adjoint with Check-pointing Strategy

Require: Checkpoint Buffer $\mathcal{C}$, Buffer size S, System Matrices $(\mathbf{A}, \mathbf{B}, \mathbf{S})$, Design variables vector $\mathbf{b}$

Ensure: Sensitivity derivatives $\nabla_{\mathbf{b}} F$

Step 1: Forward Integration of Primal System

Initialize $\mathbf{x}_{current} \leftarrow \mathbf{x}_{ss}$ using Algorithm 1

$\mathcal{C}$.advance$(0, \mathbf{x}_{ss})$ ▷ Store initial checkpoint $l = \infty$

for $k = 1$ to N_{steps} **do**

  $\mathbf{x}_{current} \leftarrow$ RK4_Primal$(\mathbf{x}_{current}, \Delta t)$

  $\mathcal{C}$.advance$(k, \mathbf{x}_{current})$ ▷ Store/Manage buffer dynamically

end for

Step 2: Initial Sensitivity Calculation ▷ Same as Algorithm 3

Step 3: Backward Integration of Adjoint System

Initialize $\mathbf{\Psi} \leftarrow \mathbf{0}$

$\mathbf{x}_{current} \leftarrow \mathcal{C}$.back().data ▷ Get final state at t_{end}

Compute $\mathbf{Q}_{current}$ from $\mathbf{x}_{current}$

for $k = N_{steps}$ **down to** 1 **do**

  $\mathcal{C}$.pop() ▷ Release last state $\mathbf{x}_k$ (no longer needed)

  // A. Retrieve Target State $\mathbf{x}_{target}$ *at* t_{k-1}

  if $\mathcal{C}$.back().index $== k - 1$ **then**

    $\mathbf{x}_{target} \leftarrow \mathcal{C}$.back().data ▷ Direct retrieval: State available

  else

    $\mathbf{x}_{temp} \leftarrow \mathcal{C}$.back().data ▷ Gap detected: Load nearest checkpoint

    $idx \leftarrow \mathcal{C}$.back().index

    while $idx < k - 1$ **do**

      $\mathbf{x}_{temp} \leftarrow$ RK4_Primal$(\mathbf{x}_{temp}, \Delta t)$

      $\mathcal{C}$.advance$(idx + 1, \mathbf{x}_{temp})$ ▷ Recompute & Refill Gap

      $idx \leftarrow idx + 1$

    end while

    $\mathbf{x}_{target} \leftarrow \mathbf{x}_{temp}@t_{k-1}$

  end if

  // B. Step Adjoint Solver $(t_k \rightarrow t_{k-1})$

  Compute $\mathbf{Q}_{target}$ from $\mathbf{x}_{target}$

  Interpolate $\mathbf{Q}(\tau) \approx (1 - \alpha)\mathbf{Q}_{current} + \alpha\mathbf{Q}_{target}$

  $\mathbf{\Psi}(t_{k-1}) \leftarrow$ RK4_Adjoint$(\mathbf{\Psi}(t_k), \mathbf{Q}_\tau, -\Delta t)$

  // C. Update (for the next step)

  $\mathbf{x}_{current} \leftarrow \mathbf{x}_{target}$

  $\mathbf{Q}_{current} \leftarrow \mathbf{Q}_{target}$

end for

Step 5: Final Assembly ▷ Same as Algorithm 3

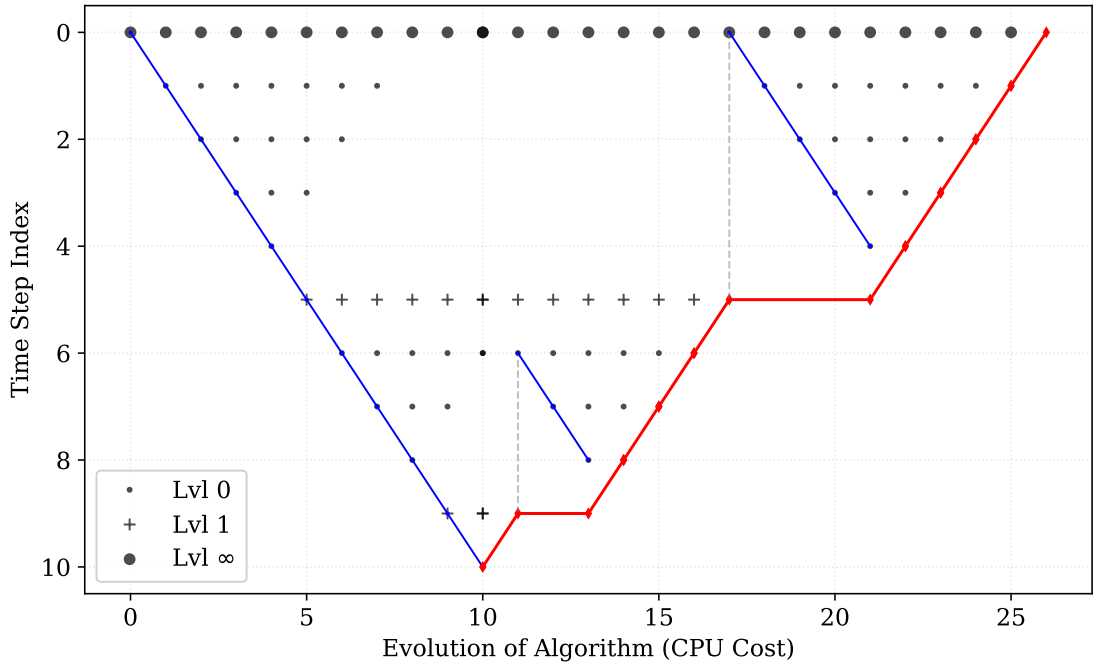


Figure 4.1: *Distribution of checkpoints for a buffer capacity of $S = 5$ over 10 time steps. Each vertical column represents the state of the buffer at a specific step in the algorithm's execution history.*

recomputation (depicted by the blue line) to reconstruct the missing intermediate states for the interval $t \in (6, 8)$. This cycle repeats until the adjoint solver reaches the initial time $t = 0$. The computational cost of the procedure is represented by the total length of the blue and red lines. The primal solver was called 16 times (initial pass + re-computations) and the adjoint solver 10 times, resulting in a total cost of 26 solver calls.

4.2.4 Performance Analysis: Computational Cost vs. Storage

To quantify the computational cost, we utilize the bound for the total number of primal recalculations n_r, derived in [17]. The total computational effort for one optimization iteration consists of one forward primal simulation followed by one backward adjoint simulation. The backward sweep requires calculating adjoint variables at every step, plus n_r re-integrations of the primal system to recover missing states from the checkpoints. Thus, the total cost in terms of simulation time equivalents

is given by:

$$C_{total} = N_{steps}(\text{Primal Steps}) + N_{steps}(\text{Adjoint Steps}) + n_r(\text{Primal Re-computations}) \tag{4.5}$$

It has been derived [17] that for a simulation with N_{steps} steps and a buffer of size S, the total recalculation count n_r is strictly bounded by the repetition number t (maximum number of times any single time step is re-calculated during the adjoint simulation):

$$n_r < t \cdot N_{steps} - \binom{S+t}{t-1} \tag{4.6}$$

For a target time discretization of $N_{steps} = 10,000$, a numerical example was conducted to measure the exact number of solver calls for different buffer sizes. The results are summarized in Table 4.1.

Table 4.1: *Impact of Buffer Size on Total Computational Cost ($N_{steps} = 10,000$)*

Buffer Size (S)	**Total Solver Calls**	**Cost Factor (C_{total}/N_{steps})**	**Regime**
10	72,918	7.29	$t = 6$
25	48,426	4.84	$t = 3$
30	48,300	4.83	$t = 3$
40	39,291	3.93	$t = 2$
50	39,296	3.93	$t = 2$
75	38,030	3.80	$t = 2$
100	34,852	3.49	$t = 2$

- **Case A: Small Buffer ($S \leq 30$).** For small buffers, the computational cost is great. For example, at $S = 10$ the algorithm required 72,918 solver calls, representing a total cost of approximately 7.3 times a single simulation. This high overhead confirms that the buffer is insufficient, forcing the algorithm into a high repetition level ($t = 6$) where states are re-calculated many times.
- **Case B: Efficient Buffer ($S = 40$).** With 40 snapshots, the total operation count dropped significantly to 39,291 solver calls. The primal solver calls in the adjoint phase are $n_r = 19,291$. The total computational cost is approximately 3.9 times that of a single primal simulation, while the memory footprint was reduced by 99.6% compared to a full-storage approach.
- **Case C: ($S \geq 50$)..** Increasing the buffer to $S = 50$ incurs a 25% memory penalty for negligible performance gain over $S = 40$. Furthermore, while $S = 100$ lowers the cost factor to 3.49×, it demands 250% of the memory required by $S = 40$ to achieve only an 11.3% improvement in simulation speed.

Based on this analysis, a buffer size of $S = 40$ is selected for $N_{steps} = 10,000$, as it represents the optimal trade-off between computational speed and memory usage.

It delivers practically identical performance to larger buffers (e.g., $S = 50$) while minimizing the storage usage. Of course, any choice of buffer size is acceptable, provided it respects the specific bounds on memory usage and computational capability imposed by the hardware infrastructure.

Chapter 5

Optimization Results

The baseline parameters for the full-vehicle model, which serve as the starting point for all optimization cases, are summarized in Table 2.1. This study adopts a **Scenario-Based Optimization** strategy. Three distinct driving scenarios are examined to identify the optimal configuration for specific environmental conditions:

1. **Case A:** Impact absorption over a deterministic bump.
2. **Case B:** Frequency response over a harmonic road profile.
3. **Case C:** Response over a stochastic road profile.

5.1 Case A: Deterministic Bump

The first case evaluates the system's transient response to a discrete obstacle, simulating a speed bump in an urban environment. The vehicle travels at a velocity of $v = 36$ km/h over a smooth cosine bump defined by Eq. 2.14, with a height of $H_{left} = 0.05\,m$ and $H_{right} = 0.06\,m$ for the left and right wheels, and length $L_{bump} = 2.5\,m$. This asymmetric profile was intentionally selected to actively excite the roll mode of the chassis. The time horizon starts at $t_{start} = 0\,s$ and ends at $t_{end} = 5\,s$. The road profile corresponds to the one depicted in Figure 2.3.

5.1.1 Optimization for a speed bump

The optimization for Case A (Deterministic Bump) achieved a 72.2% reduction in the objective function value (Table 5.1). The convergence history (Figure 5.1) shows that the ideal configuration settled after approximately 15 cycles.

When faced with a single, sharp transient event, the optimizer drove both stiffness and damping coefficients immediately to their lower bounds. This demonstrates that in the presence of a speed bump, any structural resistance is detrimental to ride comfort; soft springs are required to absorb the vertical displacement, while minimized damping prevents the rapid suspension compression from transmitting a harsh jolt to the driver.

The algorithm chose to reduce the distance a, minimizing the lever arm for the impact force on the front axle and preventing the Pitch Acceleration from building up before the rear wheels even engaged the obstacle. Also the reduction of the pitch inertia I_{yy} to its lower bound, helped to create an agile configuration. As illustrated in Figure 5.5, this setup lowers the initial peaks and smooths the overall profile of all ride comfort metrics.

Figure 5.6 plots the time history of the objective function integrand, which represents the weighted sum of the squared metrics at each time step (defined in Eq. 3.12).

Table 5.2 quantifies these improvements by presenting the percentage reduction in both maximum Peak and RMS values. The optimization was particularly effective in suppressing the initial impact, with the Peak Roll Acceleration reduced by 51.00%, its RMS reduced by 45.75% and the Peak DS Jerk by 45.99%.

Table 5.1: *Comparison of Design Variables and Objective Function for Case A*

Variable	**Baseline**	**Optimized**	**Change (%)**
Objective Function F	2.49629×10^{0}	6.95175×10^{-1}	-72.2%
K_{sf} [N/m]	1.5×10^{4}	7.5×10^{3}	-50.0%
K_{sr} [N/m]	1.7×10^{4}	8.5×10^{3}	-50.0%
B_{sf} [Ns/m]	2.5×10^{3}	1.25×10^{3}	-50.0%
B_{sr} [Ns/m]	2.5×10^{3}	1.25×10^{3}	-50.0%
I_{yy} [kg·m^2]	1.848×10^{3}	1.5×10^{3}	-18.8%
a [m]	1.20	1.04	-13.3%

Table 5.2: *Percentage Reduction in Ride Comfort Metrics for Case A (Deterministic Bump)*

Metric	**Peak Reduction (%)**	**RMS Reduction (%)**
Driver Seat Acceleration	47.47	42.56
Driver Seat Jerk	45.99	47.99
Pitch Acceleration	34.09	34.09
Roll Acceleration	51.00	45.75

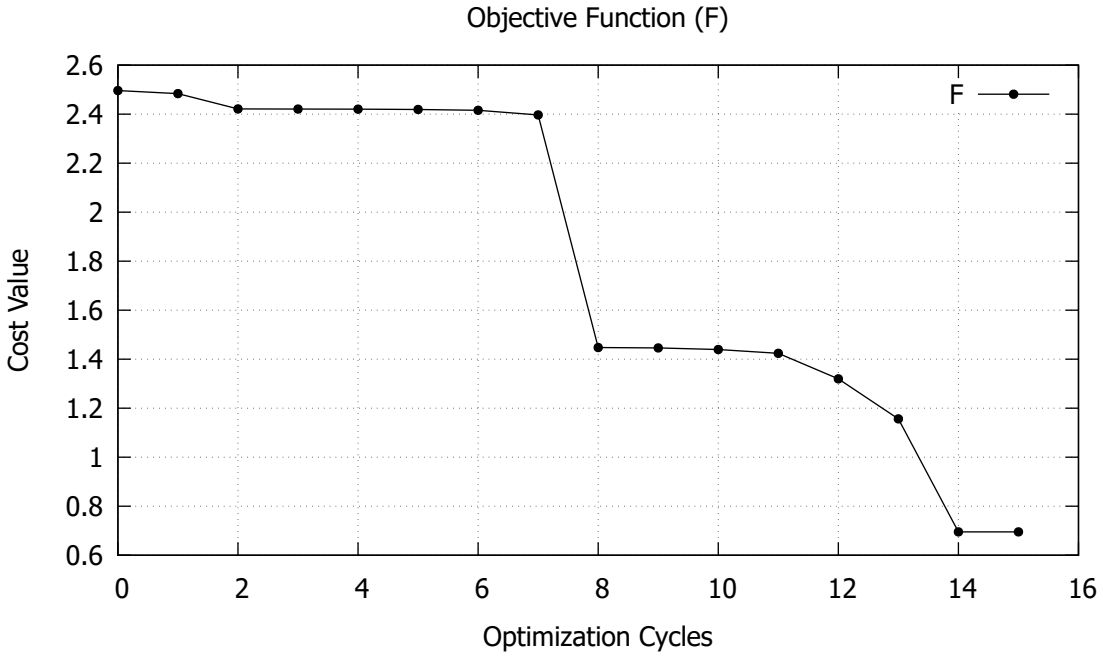


Figure 5.1: *Convergence history of the Objective Function (F) over the optimization cycles for Case A.*

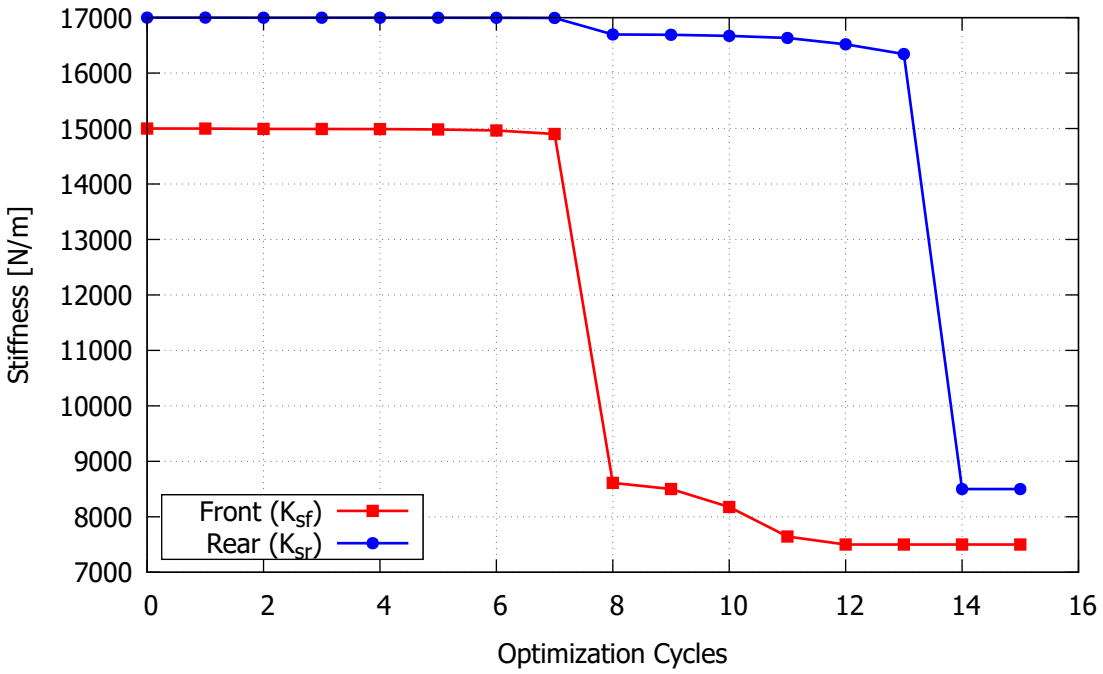


Figure 5.2: *Evolution of the Front and Rear Suspension Spring Stiffness for Case A.*

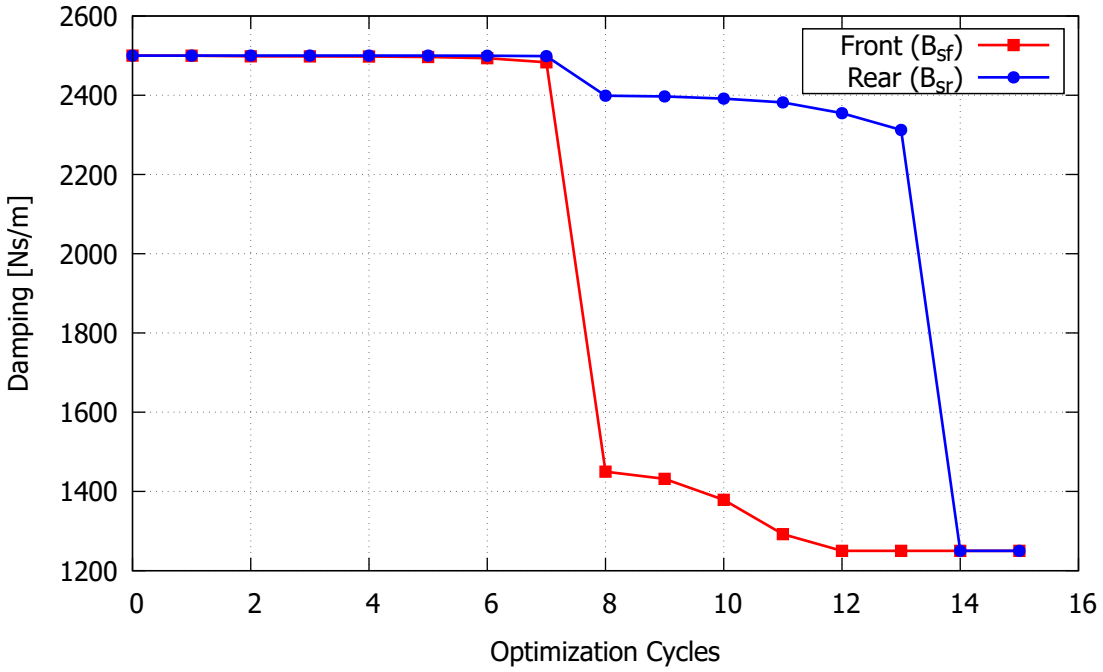


Figure 5.3: *Evolution of the Front and Rear Suspension Damping Coefficient for Case A.*

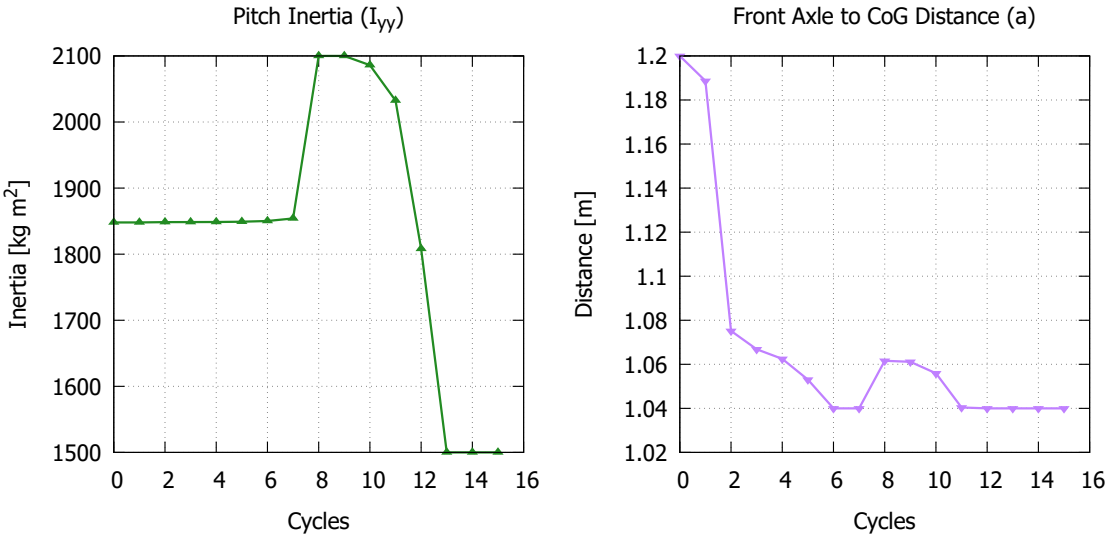


Figure 5.4: *Evolution of the Inertia and the front axle to CoG distance parameters for Case A.*

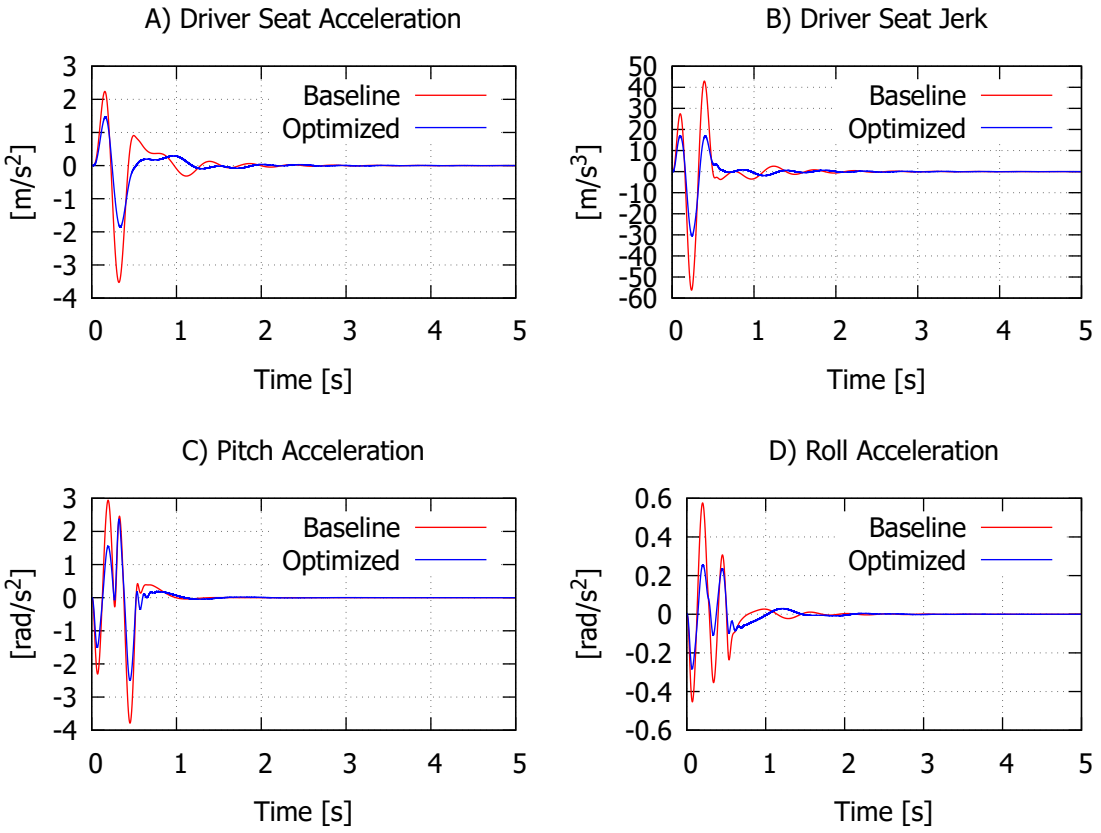


Figure 5.5: *Performance comparison of ride comfort metrics between Baseline and Optimized configurations for Case A.*

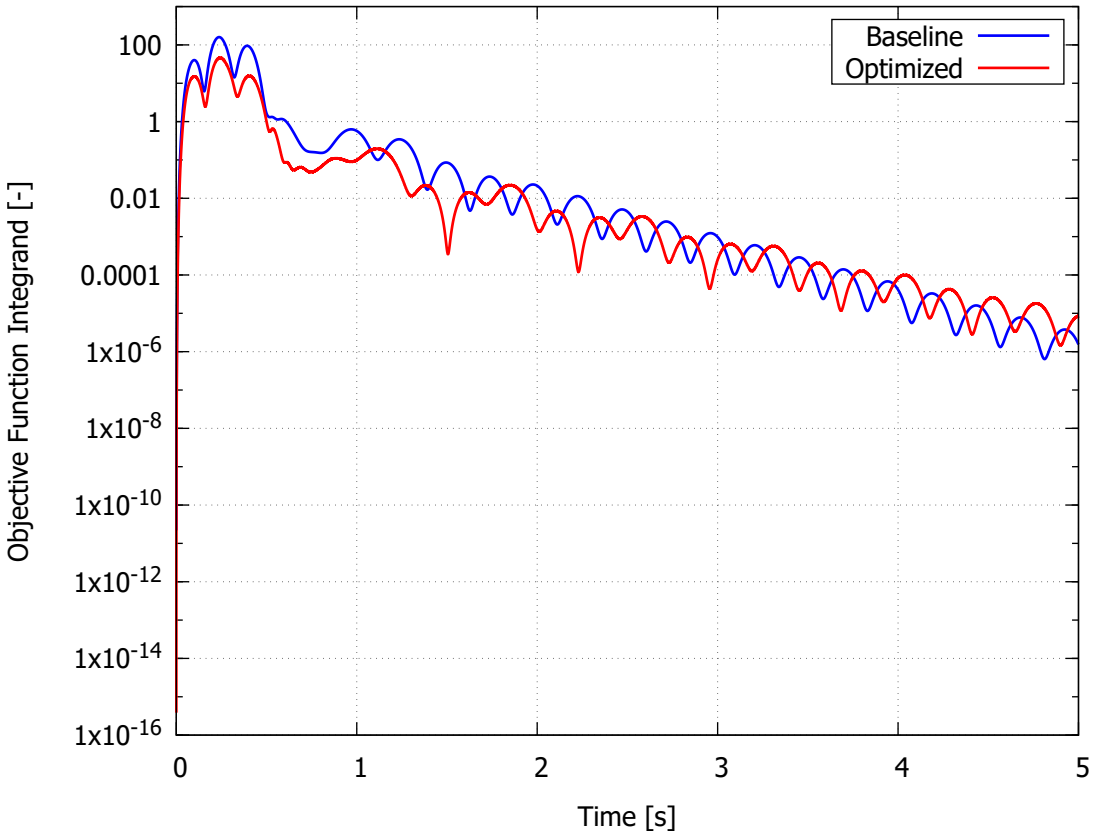


Figure 5.6: *Time evolution of the objective function integrand for Case A. The weighting coefficients that were selected:* $w_1 = w_{seat-acc} = 1.0$, $w_2 = w_{seat-jerk} = 0.05$, $w_3 = w_{pitch-acc} = 0.25$, *and* $w_4 = w_{roll-acc} = 4.0$.

5.2 Case B: Harmonic Road Profile

The second case evaluates the system's response to a harmonic excitation, simulating either unpaved roads, deformed asphalt or periodic concrete slabs. The vehicle travels at a velocity of $v = 43$ km/h over a continuous, symmetric sinusoidal road defined by Eq. 2.15, with a frequency of $\omega = 15\,rad/s$ and $A_{fl} = A_{rl} = 0.05\,m$, $A_{fr} = A_{rr} = 0.05\,m$. The time horizon starts at $t_{start} = 0\,s$ and ends at $t_{end} = 10\,s$. The road profile corresponds to the one depicted in Figure 2.4.

5.2.1 Optimization for Harmonic Road

The optimization for Case B (Harmonic Road Profile) yielded the most significant improvement among all scenarios, reducing the objective function value F by 89.0% in 29 cycles, as shown in Table 5.3. Unlike Case A, which favored agility to absorb a single impact, the results here reveal a clear strategy to stabilize the chassis against continuous forcing.

To counter the constant 15 rad/s excitation, the optimizer maximized the pitch inertia I_{yy} to its upper bound. Physically, a higher rotational inertia increases the system's resistance to the repetitive rocking motion induced by the harmonic waves. Simultaneously, the front axle to CoG distance (a) was increased to its upper limit, repositioning the axles to place the CoG further back relative to the wheelbase. This places the driver's seat closer to the midpoint of the wheelbase. In a pitching vehicle, this midpoint acts as a point of minimum vertical displacement, effectively isolating the driver from the rotational oscillation of the chassis.

While stiffness and damping were reduced, they did not hit the lower bounds seen in the transient Case A. Instead, the optimizer tuned the stiffness to ensure flexibility, while maintaining sufficient damping to strictly control the suspension motion. Notably, the optimizer maintained stiffer dampers at the rear axle ($B_{sr} > B_{sf}$) to manage the increased load at the rear. This configuration ensures that road inputs are dissipated rapidly without allowing repetitive large oscillations to develop.

Figure 5.12 shows the history of the objective function integrand. While the baseline configuration exhibits large oscillations, the optimized system effectively suppresses these fluctuations, leading to a significant reduction in the total value.

Table 5.4 reveals the effectiveness of this stabilization strategy. While the initial transient peaks were reduced by approximately 34-44%, the steady-state performance was far superior. The RMS of the Driver Seat Acceleration dropped by 70.33% and the Jerk by 73.08%.

It is observed in Figure 5.11 that the vehicle develops a roll oscillation even though the road input is symmetric. This is due to the eccentricity of the driver's seat mass ($r_y \neq 0$). As seen in Eq. 2.5, the roll moment includes the term $r_y F_{ds}$. The

symmetric road excites the heave and pitch modes, which kinematically force a vertical displacement at the point D $z_{s_{ds}}$, which is where the DS attaches to the chassis (Eq. 2.1). This creates a fluctuating suspension force F_{ds} that, due to the offset r_y, generates a time-varying moment around the roll axis.

Table 5.3: *Comparison of Design Variables and Objective Function for Case B*

Variable	**Baseline**	**Optimized**	**Change (%)**
Objective Function F	2.44198×10^{1}	2.68931×10^{0}	-89.0%
K_{sf} [N/m]	1.5×10^{4}	1.06498×10^{4}	-29.0%
K_{sr} [N/m]	1.7×10^{4}	1.06115×10^{4}	-37.6%
B_{sf} [Ns/m]	2.5×10^{3}	1.61241×10^{3}	-35.5%
B_{sr} [Ns/m]	2.5×10^{3}	2.27492×10^{3}	-9.0%
I_{yy} [kg·m^2]	1.848×10^{3}	2.1×10^{3}	13.6%
a [m]	1.20	1.43	19.2%

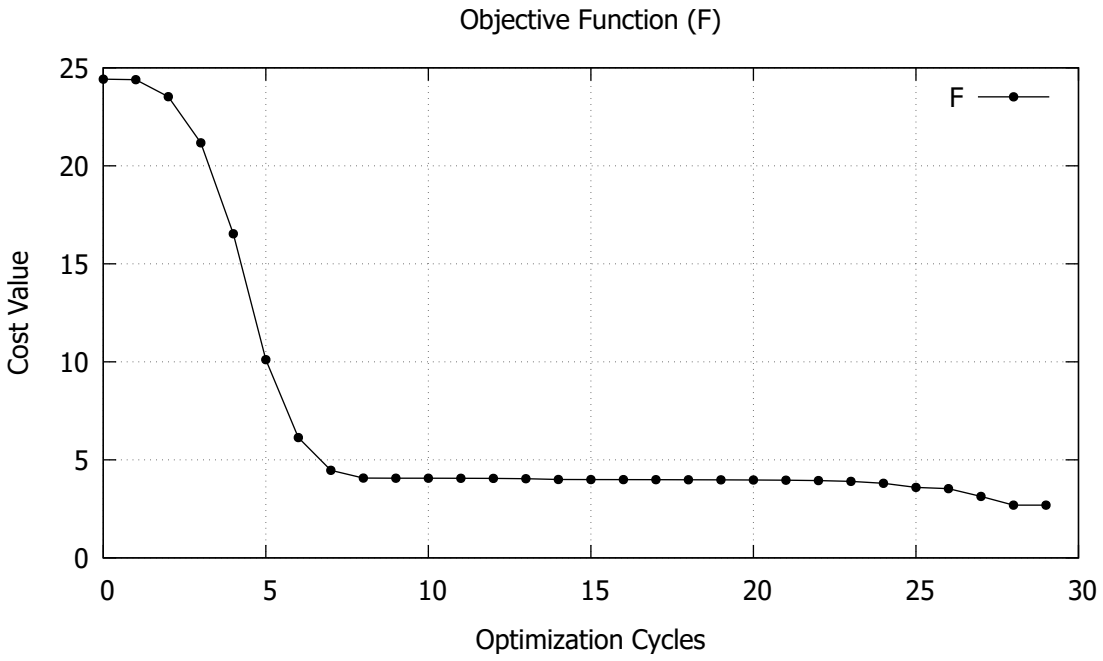


Figure 5.7: *Convergence history of the Objective Function (F) over the optimization cycles for Case B.*

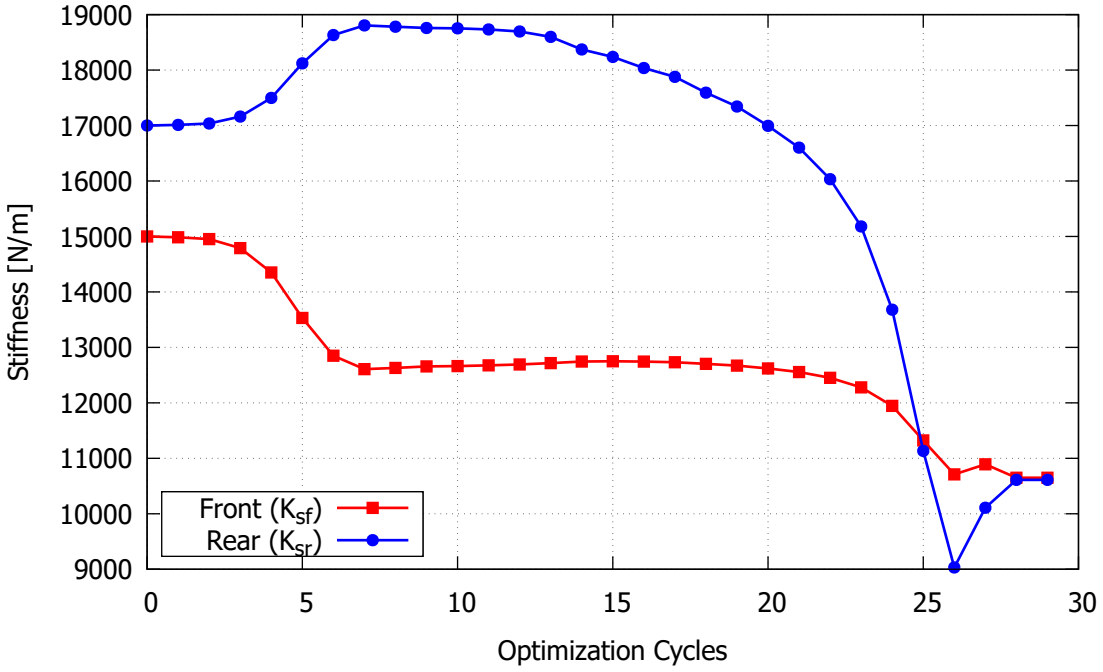


Figure 5.8: *Evolution of the Front and Rear Suspension Spring Stiffness for Case B.*

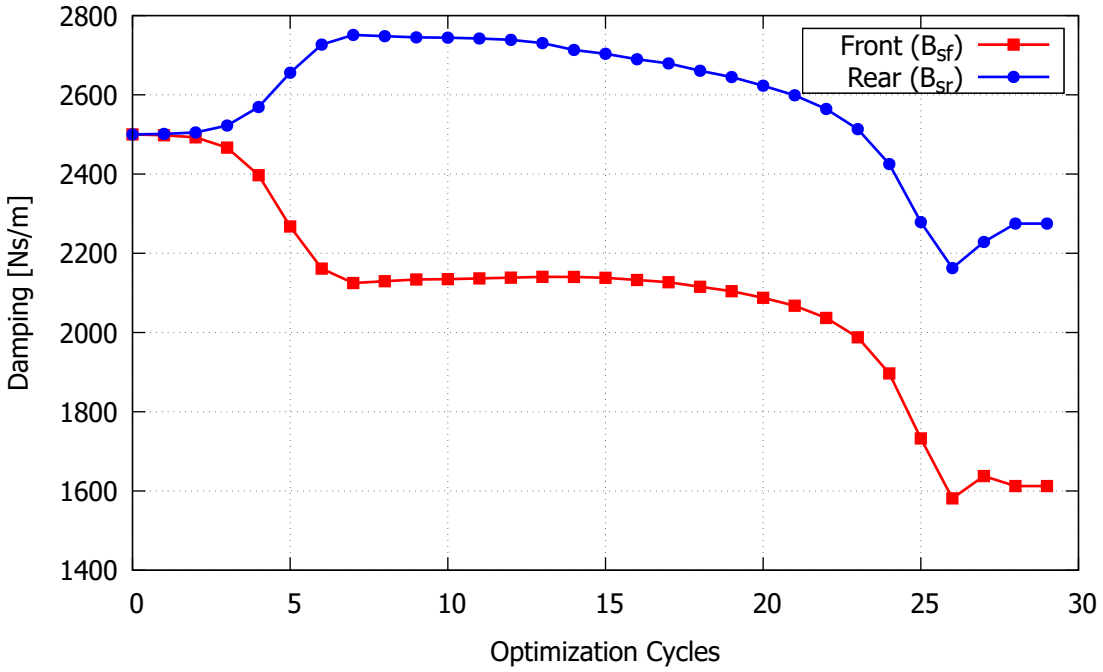


Figure 5.9: *Evolution of the Front and Rear Suspension Damping Coefficient for Case B.*

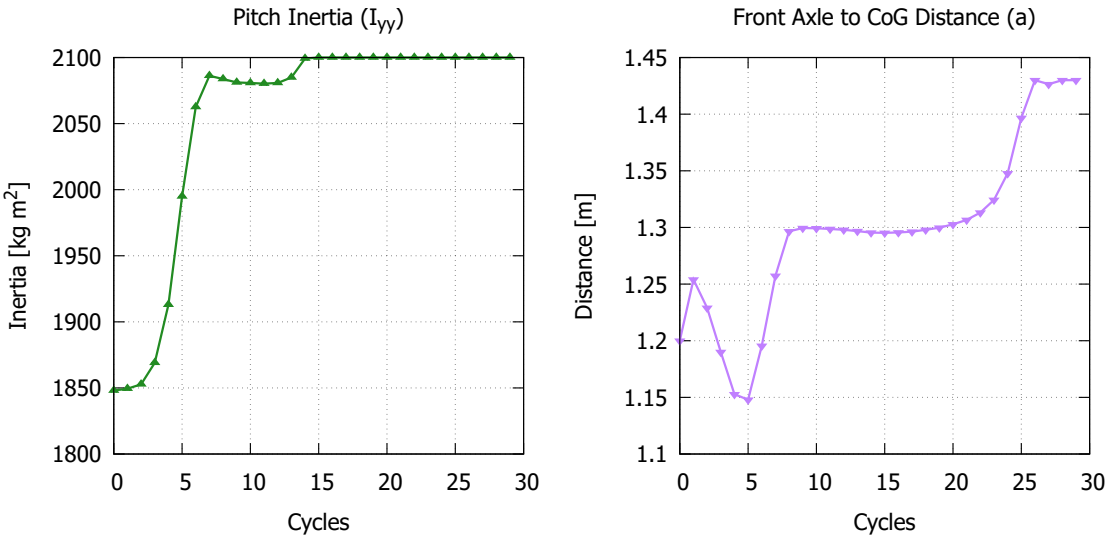


Figure 5.10: *Evolution of the Inertia and the front axle to CoG distance parameters for Case B.*

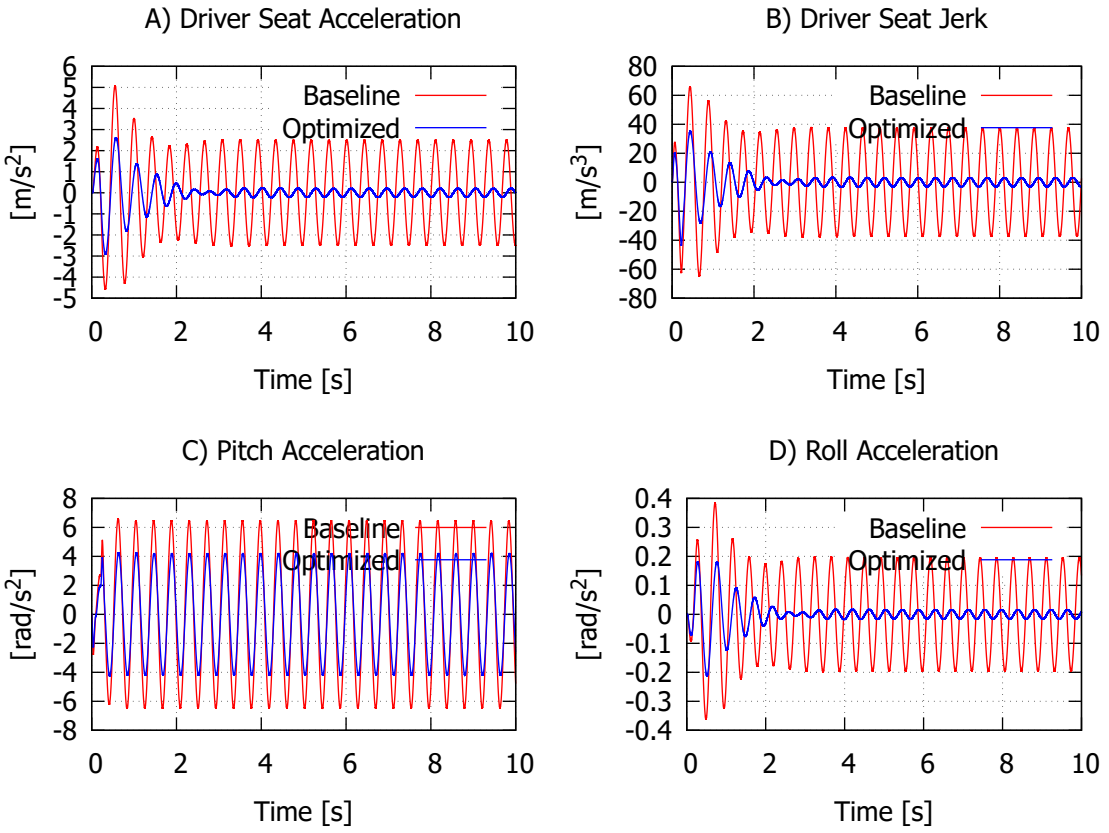


Figure 5.11: *Performance comparison of ride comfort metrics between Baseline and Optimized configurations for Case B.*

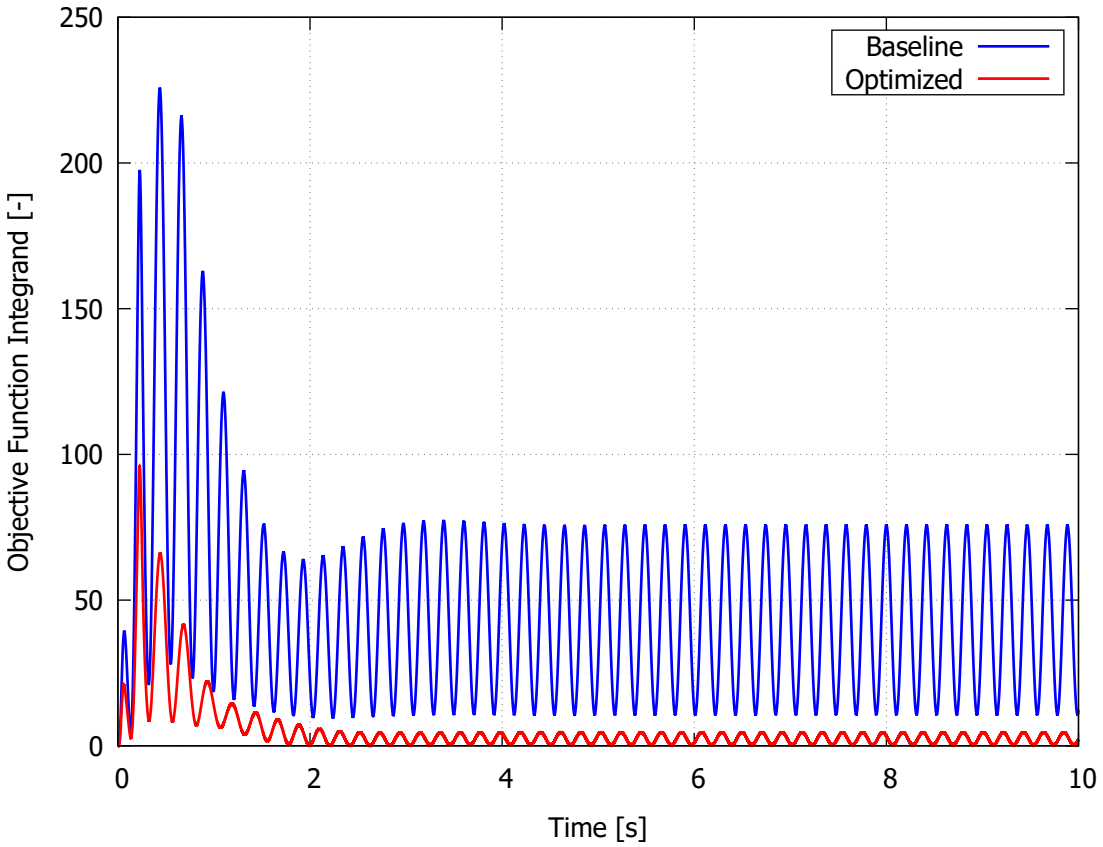


Figure 5.12: *Time evolution of the objective function integrand for Case B. The weighting coefficients that were selected:*$w_1 = w_{seat-acc} = 1.0$, $w_2 = w_{seat-jerk} = 0.05$, $w_3 = w_{pitch-acc} = 0.25$, *and* $w_4 = w_{roll-acc} = 4.0$.

Table 5.4: *Percentage Reduction in Ride Comfort Metrics for Case B (Harmonic Excitation)*

Metric	Peak Reduction (%)	RMS Reduction (%)
Driver Seat Acceleration	42.38	70.33
Driver Seat Jerk	34.10	73.08
Pitch Acceleration	35.52	34.88
Roll Acceleration	44.76	71.12

5.3 Case C: Stochastic Road Profile

The third case evaluates the system's response to a stochastic road, simulating the random surface irregularities typically found on aged asphalt or deteriorating country roads. The vehicle travels at a velocity of $v = 54$ km/h over a Class A random road generated by superpositioning $N = 500$ sinusoidal waves with Eq. 2.17. The spatial frequencies range from $n_{\min} = 0.01\,\text{cycles/m}$ to $n_{\max} = 3\,\text{cycles/m}$ and the waviness coefficient typically is $w = 2$. The time horizon starts at $t_{start} = 0\,s$ and ends at $t_{end} = 5\,s$. The road profile corresponds to the one depicted in Figure 2.5.

5.3.1 Optimization for Random Road

The optimization for Case C (Stochastic Road Profile) yielded a 62.6% reduction in the objective function (Table 5.5). The optimizer converged to a stable solution in just 4 cycles as shown in the Figure 5.13.

The front axle distance from the CoG (a) increased to its upper bound (1.43 m). This rear-biased weight distribution positions the DS near the wheelbase midpoint, trying to isolate the seat from the random pitch motions induced by the road irregularities. Additionally, the pitch inertia increased.

However, the suspension tuning reveals a front-rear bias strategy to handle the stochastic profile. The front suspension adopted a "soft spring, stiff damper" configuration. As the front wheels encounter road irregularities first, the low stiffness (K_{sf}) provides essential flexibility to filter initial impacts, while the increased damping (B_{sf}) rapidly dissipates energy to prevent post-impact oscillation. In contrast, the rear suspension kept high stiffness (K_{sr}) paired with soft damping (B_{sr}). The high stiffness strictly limits suspension travel to hold the chassis level and prevent pitch accumulation. Simultaneously, the reduced damping ensures that the rear axle does not transmit excessive force to the chassis. This split strategy allows the front to effectively isolate the driver's seat from disturbances, while the rear ensures the vehicle remains stable and planted.

Figure 5.16 shows that to minimize the global integral over time, the optimizer may accept minor momentary increases in the integrand.

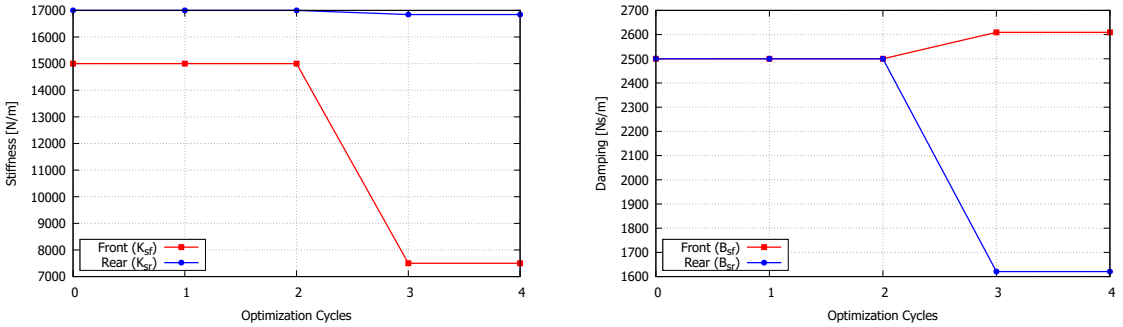
Table 5.6 details the specific improvements in ride comfort. The Pitch Acceleration proved the most difficult to suppress, showing an RMS reduction of only 9.67%. However, the strategy was highly effective for the driver, with Seat Acceleration RMS reducing by 50.03%, and Jerk by 40.90%. For the chassis, the Roll Acceleration RMS reduced by 54.85%.

Similar to Case B, a non-zero roll acceleration is observed despite the symmetric road (see Figure 5.15). This confirms that the eccentric position of the DS excites the

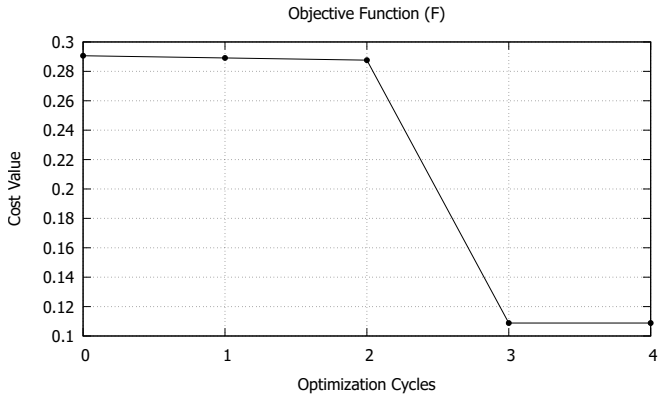
roll mode regardless of the excitation profile, following the same dynamic coupling mechanism described in Subsection 5.2.1.

Table 5.5: *Comparison of Design Variables and Objective Function for Case C*

Variable	Baseline	Optimized	Change (%)
Objective Function F	2.9062×10^{-1}	1.08809×10^{-1}	-62.6%
K_{sf} [N/m]	1.5×10^{4}	7.5×10^{3}	-50.0%
K_{sr} [N/m]	1.7×10^{4}	1.68433×10^{4}	-0.9%
B_{sf} [Ns/m]	2.5×10^{3}	2.60928×10^{3}	4.4%
B_{sr} [Ns/m]	2.5×10^{3}	1.62074×10^{3}	-35.2%
I_{yy} [kg·m^2]	1.848×10^{3}	1.92441×10^{3}	4.1%
a [m]	1.20	1.43	19.2%



(a) Spring Stiffnesses *(b) Damping Coefficients*



(c) Objective Function (F)

Figure 5.13: *Optimization results for Case C: Evolution of Stiffness (a) and Damping (b), and Convergence history of the Objective Function (c).*

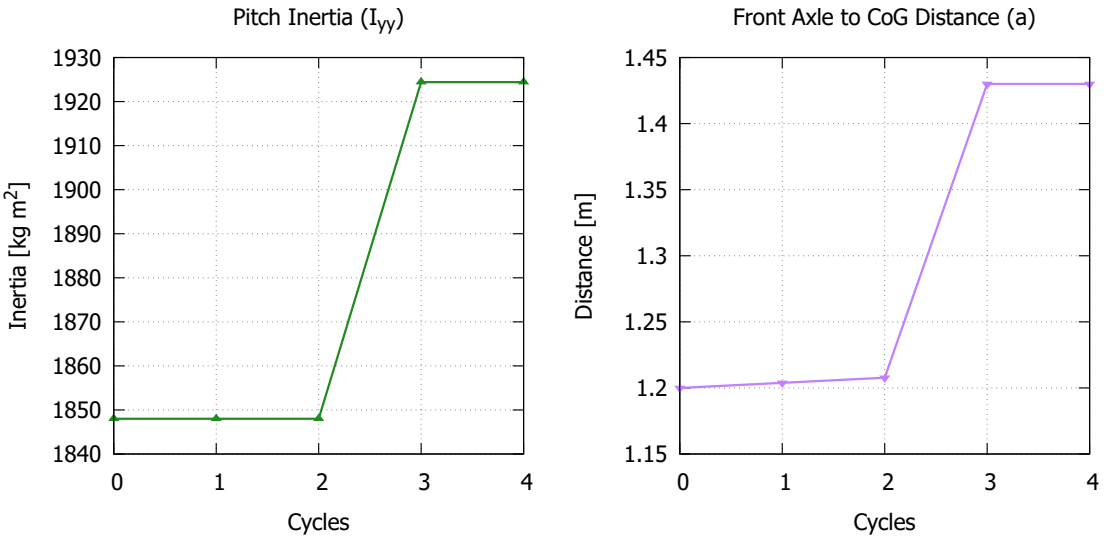


Figure 5.14: *Evolution of the Inertia and the front axle to CoG distance parameters for Case C.*

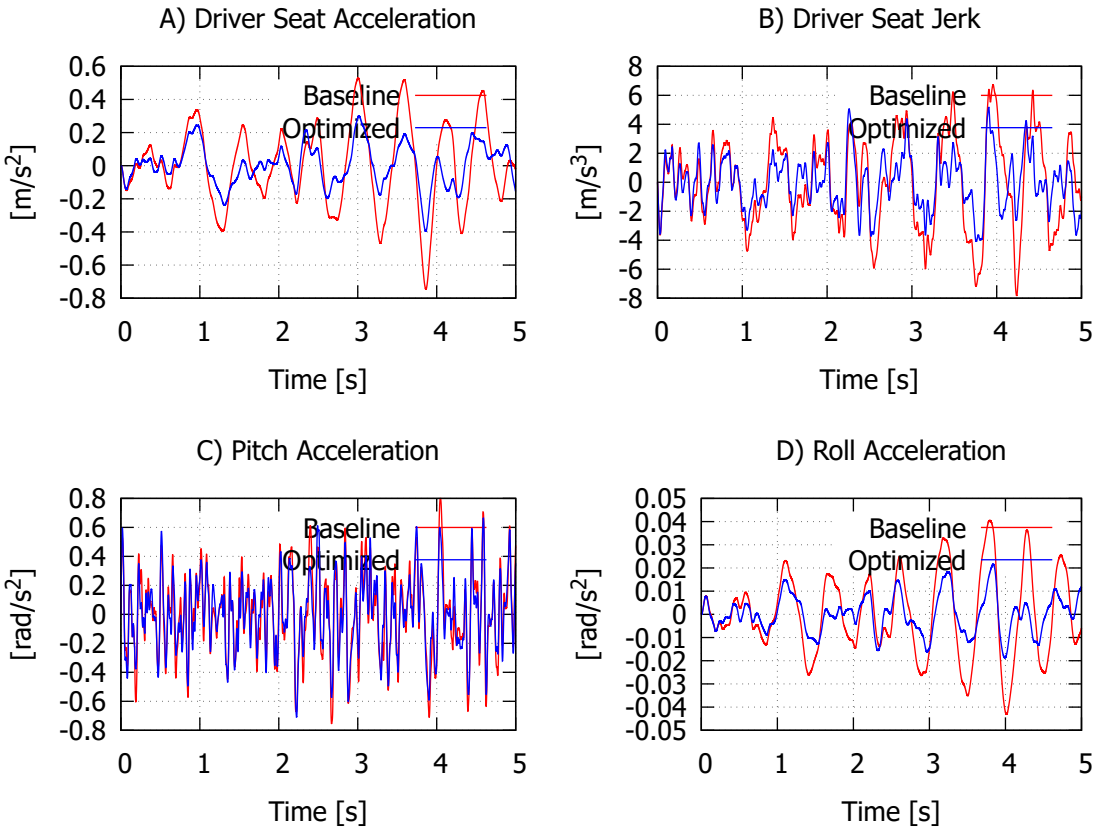


Figure 5.15: *Performance comparison of ride comfort metrics between Baseline and Optimized configurations for Case C.*

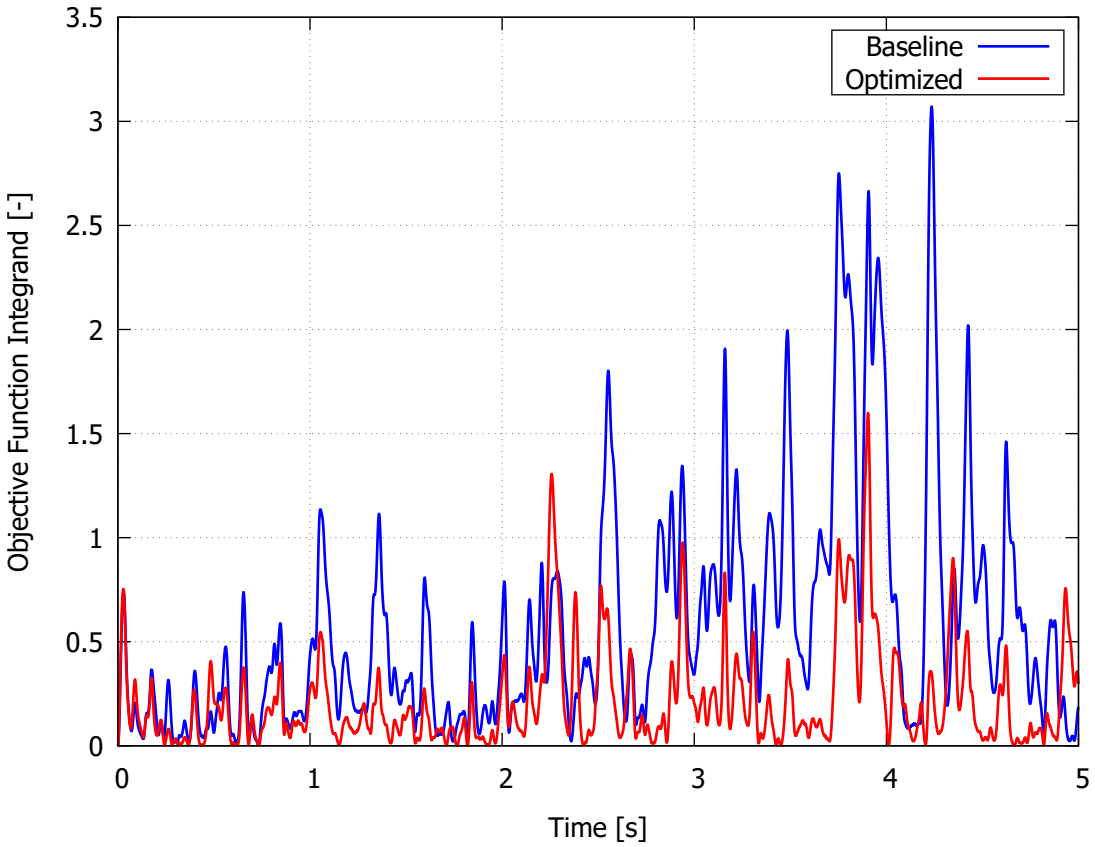


Figure 5.16: *Time evolution of the objective function integrand for Case C. The weighting coefficients that were selected:*$w_1 = w_{seat-acc} = 1.0$, $w_2 = w_{seat-jerk} = 0.05$, $w_3 = w_{pitch-acc} = 0.6$, *and* $w_4 = w_{roll-acc} = 10.0$.

Table 5.6: *Percentage Reduction in Ride Comfort Metrics for Case C (Stochastic Excitation)*

Metric	Peak Reduction (%)	RMS Reduction (%)
Driver Seat Acceleration	46.87	50.03
Driver Seat Jerk	33.95	40.90
Pitch Acceleration	11.35	9.67
Roll Acceleration	50.01	54.85

5.4 Discussion of Results

The results from the three test cases show that the optimal suspension configuration changes significantly depending on the road profile. In Case A, where the vehicle faces a single bump, the optimizer followed a strategy to absorb the impact by reducing the stiffness and the damping of the suspension, the pitch inertia and shifting the relative position of the CoG forward along the wheelbase with the DS. In Case B, where the vehicle travels in a harmonic road, the comfort was ensured by increasing the pitch inertia, shifting the relative position of the CoG rearward along with the DS, and reducing the stiffness and damping of the suspension. In Case C, where the road is random, the ride comfort resulted from a soft spring and stiff damper in the front axle, a harder spring and soft damper in the rear axle, a small increase in pitch inertia and a rear-biased weight distribution (shifting the CoG rearward relative to the axles).

It is important to note that the excitations and the general bounds of the parameters of the vehicle have been selected in order to ensure that the Modeling Limitations (Subsection 2.1.6) are not surpassed. Consequently, the linear assumptions remain valid for the specific scenarios examined. If we wanted to do more realistic environments phenomena like the wheels losing contact with the road and the existence of mechanical bump stops in the suspension that limit its travel would become vital to model in order to get meaningful and usable results.

Chapter 6

Conclusions and Future Research

This diploma thesis presented the development and implementation of an optimization framework for an 8-DoF full-vehicle suspension system using the Unsteady Continuous Adjoint Method. The core numerical solvers and the optimization algorithm were developed in C++.

A critical contribution of this work is the ride comfort objective function. While standard literature typically relies on frequency-weighted RMS acceleration (ISO 2631-1)—which requires augmenting the state vector with additional filter states—this work proposed a composite metric utilizing the vertical jerk of the Driver's Seat combined with vertical, pitch, and roll accelerations. By incorporating jerk, the objective function introduces an implicit frequency weighting where high-frequency components are penalized quadratically ($w_2\omega^2$). This captures the "harshness" of rapid acceleration changes without implementing full ISO filters.

Theoretically, the primary contribution was the derivation and application of the continuous adjoint equations for the full-vehicle model. This formulation computed the Sensitivity Derivatives with a computational cost independent of the number of design variables, and its accuracy was verified against Finite Differences.

Furthermore, the challenge of storing the primal state history in unsteady optimization was solved using the Check-pointing algorithm. A buffer size of only 40 snapshots was sufficient to reconstruct a time history of 10^4 steps. This reduced storage requirements by 99.6% while keeping the computational cost to just 3.9 times that of a single primal simulation.

Finally, the framework was validated by optimizing the vehicle under three distinct road excitations (deterministic, harmonic, and stochastic). It is important to understand that every optimized vehicle setup presented here is specifically tailored to its test case. Vehicles experience a vast variety of road profiles, and optimizing a car specifically for one scenario might lead to unwanted dynamic behavior in

another. Suspension and vehicle design, even with a single objective like ride comfort, is always a trade-off. A successful passive configuration will not be perfect for one specific situation, but will instead provide the best all-around behavior for all situations combined.

Future Research While the presented framework successfully optimizes ride comfort, the current vehicle model relies on simplifying assumptions that limit its application in extreme driving conditions. To advance this methodology, future research should focus on three key areas:

- **Integration of Non-Linearities:** The current model assumes linear tire behavior and infinite suspension travel. Future iterations should incorporate non-linear tire models and mechanical bump stops. This will require advancing the adjoint formulation to handle the non-differentiable points.
- **Multi-Objective Formulation:** The current study focused exclusively on Ride Comfort. To create a more realistic design tool, the optimization problem should be formulated as a multi-objective one, treating each comfort metric and Road Holding (safety) as separate objectives, rather than accumulating them into a single weighted objective function. This would allow for the computation of the Pareto front, revealing the true trade-offs between competing goals.
- **Application to High-Fidelity Solvers:** The Check-pointing strategy verified in this "lightweight" model should be coupled with large-scale unsteady problems, such as Computational Fluid Dynamics (CFD) or Finite Element Analysis (FEA). In these domains, where global state vectors are massive, memory-saving techniques like check-pointing are mandatory.

Bibliography

[1] Baumal, A.E., McPhee, J.J., Calamai, P.H.: Application of genetic algorithms to the design optimization of an active vehicle suspension system. Computer Methods in Applied Mechanics and Engineering **163**(1–4), 87–94 (1998)

[2] Bouazara, M., Richard, M.J.: An optimization method designed to improve 3-d vehicle comfort and road holding capability through the use of active and semi-active suspensions. European Journal of Mechanics - A/Solids **20**(3), 509–520 (2001). https://doi.org/https://doi.org/10.1016/S0997-7538(01)01138-X, https://www.sciencedirect.com/science/article/pii/S099775380101138X

[3] Fossati, G.G., Miguel, L.F.F., Casas, W.J.P.: Multi-objective optimization of the suspension system parameters of a full vehicle model. Optimization and Engineering **20**, 151 – 177 (2018), https://api.semanticscholar.org/CorpusID:125555627

[4] Giannakoglou, K.: Optimization Methods in Aerodynamics. TNUA (2006)

[5] Griewank, A., Walther, A.: Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. ACM Transactions on Mathematical Software **26**(1), 19–45 (2000)

[6] Griffin, M.J.: Handbook of Human Vibration. Academic Press (1990)

[7] Hu, S., Rui, X., Gu, J., Zhang, X.: Dynamics modeling and suspension parameters optimization of vehicle system based on reduced multibody system transfer matrix method. Machines **13**(2), 116 (Feb 2025). https://doi.org/10.3390/machines13020116, http://dx.doi.org/10.3390/machines13020116

[8] Huang, Q., Wang, H.: Fundamental study of jerk: Evaluation of shift quality and ride comfort. In: SAE Technical Paper Series. vol. 1. SAE International, SAE International (2004)

[9] International Organization for Standardization: ISO 2631-1:1997: Mechanical vibration and shock – evaluation of human exposure to whole-body vibration – part 1: General requirements (1997)

[10] International Organization for Standardization: ISO 8608:2016: Mechanical vibration – road surface profiles – reporting of measured data (2016)

[11] Jazar, R.N.: Vehicle Dynamics: Theory and Application. Springer US, Boston, MA (2008)

[12] Khajavi, M., Notghi, B., Paygane, G.: Multi objective optimization approach to optimize vehicle ride and handling characteristics **62**, 580–584 (02 2010)

[13] Margetis, A.S., Papoutsis-Kiachagias, E., Giannakoglou, K.: Lossy compression techniques supporting unsteady adjoint on 2d/3d unstructured grids. Computer Methods in Applied Mechanics and Engineering **387**, 114152 (2021)

[14] Meng, R., Xie, N., Wang, L.: Multiobjective game method based on self-adaptive space division of design variables and its application to vehicle suspension. Mathematical Problems in Engineering **2014**(1) (Jan 2014). https://doi.org/10.1155/2014/479272, http://dx.doi.org/10.1155/2014/479272

[15] Mitra, A.C., Kiranchand, G.R., Soni, T., Banerjee, N.: Optimization of passive vehicle suspension system by genetic algorithm. Procedia Engineering **144**, 1158–1166 (2016). https://doi.org/10.1016/j.proeng.2016.05.087

[16] Mitra, A., Benerjee, N., Khalane, H.A., Sonawane, M.A., Joshi, D.R., Bagul, G.: Simulation and analysis of full car model for various road profile on a analytically validated matlab / simulink model (2013), https://api.semanticscholar.org/CorpusID:1555339

[17] Moin, P., Wang, Q., Iaccarino, G.: Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation. SIAM Journal on Scientific Computing **31**(4), 2549–2567 (2009)

[18] Mudduluru, S.R., Chizari, M.: Quarter and full car models optimisation of passive and active suspension system using genetic algorithm. arXiv preprint, arXiv:2101.12629 (2021)

[19] Nocedal, J., Wright, S.J.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering, Springer, New York, NY, 2nd edn. (2006). https://doi.org/10.1007/978-0-387-40065-5

[20] Rajamani, R.: Vehicle Dynamics and Control. Springer Science & Business Media (2011)

[21] RIMELL, A.N., MANSFIELD, N.J.: Design of digital filters for frequency weightings required for risk assessments of workers exposed to vibration. Industrial Health **45**(4), 512–519 (2007). https://doi.org/10.2486/indhealth.45.512, http://dx.doi.org/10.2486/indhealth.45.512

[22] Sayers, M.W., Karamihas, S.M.: The little book of profiling: Basic information about measuring and interpreting road profiles. Tech. Rep. UMTRI-98-38, University of Michigan Transportation Research Institute, Ann Arbor, MI (1998)

[23] Shirahatti, A., Prasad, P., Panzade, P., Kulkarni, M.: Optimal design of passenger car suspension for ride and road holding. Journal of The Brazilian Society of Mechanical Sciences and Engineering - J BRAZ SOC MECH SCI ENG **30** (03 2008). https://doi.org/10.1590/S1678-58782008000100010

[24] Thoresson, M.J.: Efficient Gradient-Based Optimisation of Suspension Characteristics for an Off-Road Vehicle. Phd thesis, University of Pretoria, Pretoria, South Africa (July 2007)

[25] Tyan, F., Hong, Y.F., Tu, S.H., Jeng, W.S.: Generation of random road profiles. Journal of the Chinese Society of Mechanical Engineers **30**(2), 137–142 (2009)

[26] Verros, G., Natsiavas, S., Papadimitriou, C.: Design optimization of quarter-car models with passive and semi-active suspensions under random road excitation. Journal of Vibration and Control **11**(5), 581—606 (2005)

[27] Vezyris, C.K., Kavvadias, I.S., Papoutsis-Kiachagias, E.M., Giannakoglou, K.C.: Unsteady continuous adjoint method using pod for jet-based flow control. In: 11th World Congress on Computational Mechanics (WCCM XI). Barcelona, Spain (2014)

[28] Vezyris, C.K., Papoutsis-Kiachagias, E.M., Giannakoglou, K.C.: On the incremental singular value decomposition method to support unsteady adjoint based optimization. International Journal for Numerical Methods in Fluids **91**(7), 315–331 (2019)

[29] Walton, S., Hassan, O., Morgan, K.: Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions. Applied Mathematical Modelling **37**(20), 8930–8945 (2013)

[30] Wang, C.Y., Xu, Z.J., Zhao, W.Z.: Parameter optimization of automotive suspension system based on particle swarm algorithm. WSEAS Transactions on Systems and Control **10**, 608–615 (2015)

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Μηχανολογων Μηχανικων
Τομεας Ρευστων
Μοναδα Παραλληλης Υπολογιστικης
Ρευστοδυναμικης & Βελτιστοποιησης

Βελτιστοποίηση Συστήματος Ανάρτησης Πλήρους Οχήματος με Χρήση της Χρονικά Μη-Μόνιμης Συνεχούς Συζυγούς Μεθόδου

Εκτενής Περίληψη Διπλωματικής Εργασίας

Δημήτριος Μαυρομάτης

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

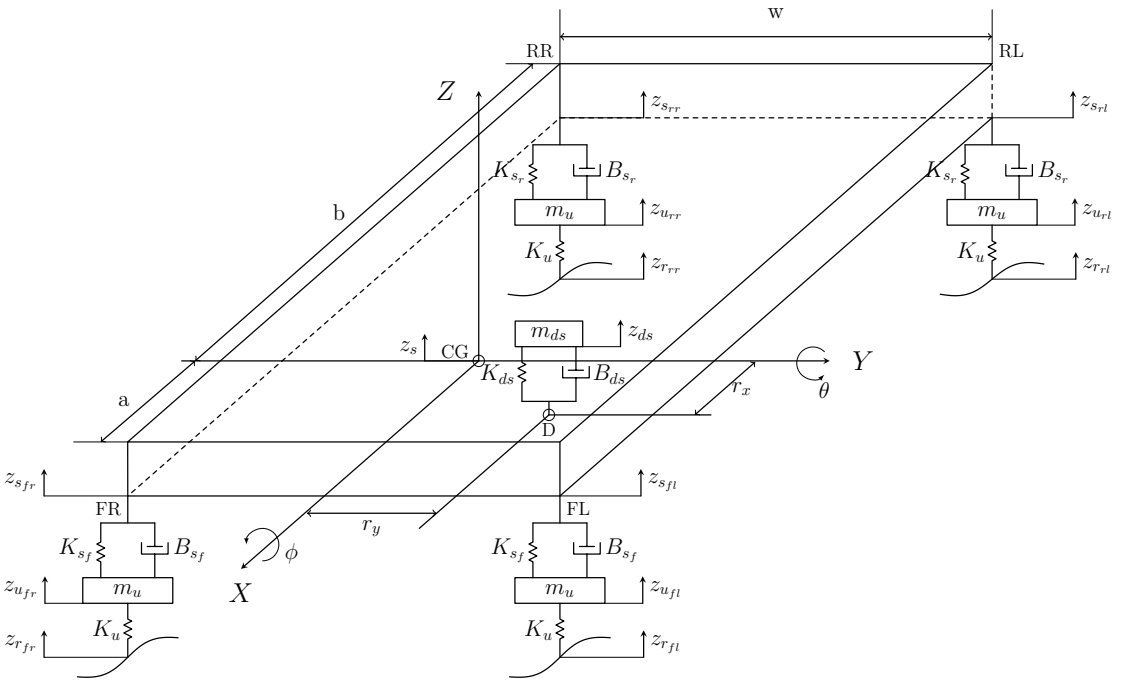
Αθήνα, 2026

Εισαγωγή

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου πλαισίου βελτιστοποίησης για το σύστημα ανάρτησης οχήματος, με στόχο τη μεγιστοποίηση της άνεσης των επιβατών. Η καινοτομία της μελέτης έγκειται στην εφαρμογή της μη μόνιμης συνεχούς συζυγούς μεθόδου (unsteady continuous adjoint method) σε ένα μοντέλο ολόκληρου οχήματος (full-car model), επιτρέποντας τον υπολογισμό των παραγώγων ευαισθησίας με υπολογιστικό κόστος ανεξάρτητο από τον αριθμό των μεταβλητών σχεδιασμού. Για την αντιμετώπιση των υψηλών απαιτήσεων σε μνήμη που συνεπάγεται η συζυγής μέθοδος στο πεδίο του χρόνου, εφαρμόζεται η τεχνική του Check-pointing, η οποία επιτυγχάνει μια βέλτιστη ισορροπία μεταξύ χρήσης μνήμης και υπολογιστικού χρόνου.

Μοντελοποίηση Δυναμικής Οχήματος

Για την ανάλυση της κατακόρυφης δυναμικής, χρησιμοποιείται ένα γραμμικό μοντέλο 8 βαθμών ελευθερίας (DoF). Το σύστημα συντεταγμένων ορίζεται έτσι ώστε ο άξονας X



Σχήμα 6.1: *Μοντέλο ανάρτησης πλήρους οχήματος με κάθισμα.*

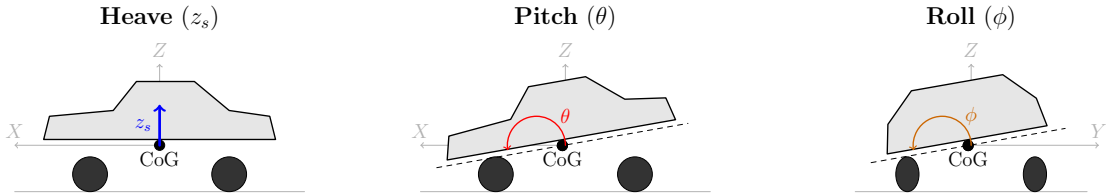
να συμπίπτει με τη διαμήκη διεύθυνση του οχήματος (με κατεύθυνση προς τα εμπρός), ο άξονας Y με την εγκάρσια και ο άξονας Z με την κατακόρυφη διεύθυνση. Η αρχή των αξόνων τοποθετείται στο κέντρο βάρους του οχήματος, στην κατάσταση όπου οι αναρτήσεις είναι αφόρτιστες. Το μοντέλο του οχήματος συντίθεται από τα ακόλουθα

επιμέρους μοντέλα μαζών, η σύνθεσή τους φαίνεται στο Σχήμα 6.1. Επίσης, αυτό το σύστημα χρησιμοποιείται συχνά στη βιβλιογραφία [12, 16, 23, 2, 3]:

Αμάξωμα (Αναρτώμενη μάζα): Το αμάξωμα (σασί) μοντελοποιείται ως μια συμπαγής αναρτώμενη μάζα m_s. Το αμάξωμα έχει 3 βαθμούς ελευθερίας:

- Την κατακόρυφη μετατόπιση (z_s) του Κέντρου Βάρους
- Την πρόνευση (θ), που είναι η περιστροφή του αμαξώματος γύρω από τον άξονα Y)
- Την πλευρική κλίση (ϕ), που είναι η περιστροφή γύρω από τον άξονα X).

Η Εικόνα 6.2 δείχνει τους παραπάνω 3 βαθμούς ελευθερίας του αμαξώματος.



Σχήμα 6.2: *Αναπαράσταση των βαθμών ελευθερίας του αμαξώματος: Κατακόρυφη Μετατόπιση (Heave), Πρόνευση (Pitch) και Πλευρική Κλίση (Roll).*

Τροχοί (Μη αναρτώμενες μάζες): Οι τροχοί (μη αναρτώμενες μάζες m_u) συνδέονται με το σασί μέσω αναρτήσεων που περιλαμβάνουν ένα ελατήριο και έναν αποσβεστήρα με σταθερές K_s και B_s αντίστοιχα. Το όχημα περιλαμβάνει ανεξάρτητη ανάρτηση σε κάθε τροχό αλλά οι σταθερές μένουν ίδιες και για τις δυο αναρτήσεις του εκάστοτε άξονα. Ο κάθε τροχός είναι σχεδιασμένος για να κινείται κατακόρυφα, άρα έχει 1 βαθμό ελευθερίας (κατακόρυφη μετατόπιση z_u), συνολικα 4 βαθμοί ελευθερίας.

Κάθισμα Οδηγού: Ένα μοντέλο καθίσματος οδηγού με μάζα m_{ds}, συνδέεται πάνω στο σασί με τη δική του ανάρτηση (K_{ds} και B_{ds}). Έχει 1 βαθμό ελευθερίας (κατακόρυφη μετατόπιση z_{ds}) ο οποίος χρησιμεύει στην ακριβή αξιολόγηση της επιτάχυνσης που δέχεται ο οδηγός.

Η αρχική επιλογή για τις παραμέτρους που χαρακτηρίζουν το όχημα παρουσιάζονται στον Πίνακα 6.1. Περιέχει επίσης τα όρια ορισμένων τιμών όπως αυτά χρησιμοποιήθηκαν στη βιβλιογραφία [14, 3].

Η κίνηση των σημείων στήριξης της ανάρτησης στα άκρα του αμαξώματος και στο σημείο έδρασης του καθίσματος προκύπτει από την υπέρθεση των κινήσεων κατακόρυφης μετατόπισης, πρόνευσης και πλευρικής κλίσης του σασί. Οι δυνάμεις της ανάρτησης μοντελοποιούνται ως γραμμικές, προερχόμενες από τη σχετική μετατόπιση και ταχύτητα μεταξύ των άκρων του σασί και των τροχών. Αντίστοιχα, η δύναμη στο κάθισμα του οδηγού προκύπτει από τη σχετική κίνηση μεταξύ της μάζας του καθίσματος και του σημείου έδρασής του στο σασί.

Πίνακας 6.1: *Παράμετροι Αρχικού Μοντέλου Πλήρους Οχήματος (Πριν τη Βελτιστοποίηση)*

Παράμετρος	**Σύμβολο**	**Τιμή**	**Μονάδα**
Σύστημα Ανάρτησης			
Ακαμψία Εμπρός	K_{sf}	$7{,}500 \leq 15{,}000 \leq 22{,}500$	N/m
Ακαμψία Πίσω	K_{sr}	$8{,}500 \leq 17{,}000 \leq 25{,}500$	N/m
Απόσβεση Εμπρός	B_{sf}	$1{,}250 \leq 2{,}500 \leq 3{,}750$	Ns/m
Απόσβεση Πίσω	B_{sr}	$1{,}250 \leq 2{,}500 \leq 3{,}750$	Ns/m
Μη Αναρτώμενη Μάζα (τροχός)	m_u	40	kg
Ακαμψία Ελαστικού	K_u	250,000	N/m
Κάθισμα Οδηγού			
Θέση Καθίσματος ως προς το Κ.Β. (x)	r_x	0.2	m
Θέση Καθίσματος ως προς το Κ.Β. (y)	r_y	0.375	m
Ακαμψία Καθίσματος	K_{ds}	15,000	N/m
Απόσβεση Καθίσματος	B_{ds}	150	Ns/m
Μάζα Οδηγού και Καθίσματος	m_{ds}	100	kg
Πλαίσιο (Σασί)			
Μεταξόνιο	l	2.6	m
Απόσταση Εμπρός άξονα από Κ.Β.	a	$1.04 \leq 1.2 \leq 1.43$	m
Απόσταση Πίσω άξονα από Κ.Β.	b	$1.17 \leq 1.4 \leq 1.56$	m
Μετατρόχιο	w	1.5	m
Μάζα Πλαισίου	m_s	1,100	kg
Ροπή Αδράνειας Πλευρικής Κλίσης	I_{xx}	550	kg·m^2
Ροπή Αδράνειας Πρόνευσης	I_{yy}	$1{,}500 \leq 1{,}848 \leq 2{,}100$	kg·m^2

Οι διαφορικές εξισώσεις που διέπουν το σύστημα προκύπτουν από την εφαρμογή του δεύτερου νόμου του Νεύτωνα. Για την αναρτώμενη μάζα, λαμβάνεται η ισορροπία των δυνάμεων στον κατακόρυφο άξονα και η ισορροπία ροπών για τις περιστροφές.

$$m_s \ddot{z}_s = F_{ds} - W_s - (F_{s_{fl}} + F_{s_{fr}} + F_{s_{rl}} + F_{s_{rr}}) \tag{6.1}$$

$$I_{yy} \ddot{\theta} = -r_x F_{ds} + a(F_{s_{fl}} + F_{s_{fr}}) + b(-F_{s_{rl}} - F_{s_{rr}}) \tag{6.2}$$

$$I_{xx} \ddot{\phi} = r_y F_{ds} + \frac{w}{2}(-F_{s_{fl}} - F_{s_{rl}}) + \frac{w}{2}(F_{s_{fr}} + F_{s_{rr}}) \tag{6.3}$$

Για τις μη αναρτώμενες μάζες, η κίνηση κάθε τροχού καθορίζεται από τη δύναμη της ανάρτησης και την αντίδραση του ελαστικού, το οποίο προσομοιώνεται ως γραμμικό ελατήριο ακαμψίας K_u σε επαφή με το προφίλ του δρόμου.

$$m_u \ddot{z}_{u_{fl}} = F_{s_{fl}} - W_u - K_u(z_{u_{fl}} - z_{r_{fl}}) \tag{6.4}$$

Για το κάθισμα του οδηγού λαμβάνεται η ισορροπία των δυνάμεων στον κατακόρυφο άξονα.

$$m_{ds}\ddot{z}_{ds} = -F_{ds} - W_{ds} \tag{6.5}$$

Οι παραπάνω 8 εξισώσεις συντάσσουν ένα σύστημα 16 διαφορικών εξισώσεων πρώτης τάξης. Το **Πρωτεύον Σύστημα** είναι γραμμικό και χρονικά αμετάβλητο σύστημα και έχει την εξής μορφή:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{S} \tag{6.6}$$

Όπου Το διάνυσμα κατάστασης $\mathbf{x} \in \mathbb{R}^{16}$ αποτελείται από τις μετατοπίσεις και τις ταχύτητες των 8 μεταβλητών:

$$\mathbf{x} = [z_s, \dot{z}_s, \theta, \dot{\theta}, \phi, \dot{\phi}, \underbrace{z_{u_{fl}}, \dot{z}_{u_{fl}}, \ldots}_{\text{4 wheels}}, z_{ds}, \dot{z}_{ds}]^T \tag{6.7}$$

Οι διεγέρσεις του οδοστρώματος αποτελούν το διάνυσμα $\mathbf{u}(t) \in \mathbb{R}^4$, που περιέχει το προφίλ του εκάστοτε δρόμου ως συνάρτηση του χρόνου σε κάθε επαφή του με τους τροχούς:

$$\mathbf{u}(t) = [z_{r_{fl}}(t), z_{r_{fr}}(t), z_{r_{rl}}(t), z_{r_{rr}}(t)]^T \tag{6.8}$$

Το διάνυσμα πηγής $\mathbf{S} \in \mathbb{R}^{16}$ περιέχει τους όρους με τις επιταχύνσεις της βαρύτητας $-g$. Ο πίνακας του συστήματος $\mathbf{A} \in \mathbb{R}^{16\times16}$ και ο Πίνακας εισόδου $\mathbf{B} \in \mathbb{R}^{16\times4}$. Το μοντέλου του οχήματος έχει ορισμένους περιορισμούς όπως η αδυναμία μοντελοποίησης της αποκόλλησης του τροχού από το έδαφος, μη ύπαρξη φυσικών ορίων στις αναρτήσεις, μικρές γωνίες πλευρικής κλίσης και πρόνευσης. Η αριθμητική επίλυση του πρωτεύοντος προβλήματος πραγματοποιείται με τη μέθοδο Runge-Kutta 4ης τάξης, αφού προηγηθεί η επίλυση του προβλήματος στατικής ισορροπίας με τη Μέθοδο Απαλοιφής Gauss.

Βελτιστοποίηση και η Μη Μόνιμη Συζυγής Μέθοδος

Η συνάρτηση κόστους που θα χρησιμοποιηθεί αποτελείται από το σταθμισμένο τετραγωνισμένο άθροισμα των δεικτών άνεσης. Αυτοί είναι οι επιταχύνσεις του Καθίσματος του Οδηγού $\dot{x}_{16}$, της πρόνευσης $\dot{x}_4$ και της πλευρικής κλίσης x_6 του αμαξώματος και η χρονική μεταβολή της επιτάχυνσης του Καθίσματος του Οδηγού $\ddot{x}_{16}$:

$$F = \frac{1}{2\,t_{period}} \int_{t_{start}}^{t_{end}} \left(w_1\dot{x}_{16}^2 + w_2\ddot{x}_{16}^2 + w_3\dot{x}_4^2 + w_4\dot{x}_6^2\right) dt, \tag{6.9}$$

Όπου $w_1, \ldots, w_4$ είναι οι συντελεστές βαρύτητας και $t_{period} = t_{end} - t_{start}$ είναι η χρονική διάρκεια του ολοκληρώματος. Ορίζεται επίσης το διάνυσμα των μεταβλητών

σχεδιασμού:

$$\mathbf{b} = \begin{bmatrix} K_{s_f} & K_{s_r} & B_{s_f} & B_{s_r} & I_{yy} & a \end{bmatrix}^T \tag{6.10}$$

Για τον υπολογισμό της κλίσης της F ως προς τις μεταβλητές σχεδιασμού (π.χ. ακαμψίες ελατηρίων, συντελεστές απόσβεσης), χρησιμοποιείται η Συζυγής Μέθοδος. Ορίζεται η επαυξημένη συνάρτηση κόστους:

$$F_{aug} = F + \int_{t_{start}}^{t_{end}} \mathbf{\Psi}^T \mathbf{R} dt = \frac{1}{2\, t_{period}} \int_{t_{start}}^{t_{end}} \mathcal{L}(\mathbf{x}, \mathbf{u}, \mathbf{b}) dt + \int_{t_{start}}^{t_{end}} \mathbf{\Psi}^T \left(\dot{\mathbf{x}} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u} - \mathbf{S} \right) dt \tag{6.11}$$

όπου $\mathcal{L}$ αντιπροσωπεύει των σταθμισμένο άθροισμα των τετραγώνων των τεσσάρων δεικτών άνεσης και $\mathbf{\Psi} \in \mathbb{R}^{16\times 1}$ είναι το διάνυσμα των συζυγών μεταβλητών κατάστασης. Η εξίσωση διαφοροποιείται σε σχέση με το διάνυσμα των μεταβλητών σχεδιασμού. Μέσω ολοκλήρωσης κατά μέρη και εφαρμογής του κανόνα του Leibniz, προκύπτουν οι διαφορικές εξισώσεις του **Συζυγούς Συστήματος**:

$$\frac{d\mathbf{\Psi}^T}{dt} = -\mathbf{\Psi}^T \mathbf{A} + \frac{1}{2t_{period}} \mathbf{Q}(t) \tag{6.12}$$

Όπου ο όρος $\mathbf{Q}(t)$ είναι το διάνυσμα που αποτελείται από τους δείκτες άνεσης της συνάρτησης κόστους και αποτελείται από τα εξής μέλη:

$$\mathbf{Q_{seat-acc}} = 2w_1(\mathbf{A_{16}}\mathbf{x} + S_{16})\mathbf{A_{16}} \tag{6.13α′}$$

$$\mathbf{Q_{seat-jerk}} = 2w_2[\mathbf{A_{16}}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{S})]\mathbf{A_{16}}\mathbf{A} \tag{6.13β′}$$

$$\mathbf{Q_{pitch-acc}} = 2w_3(\mathbf{A_4}\mathbf{x})\mathbf{A_4} \tag{6.13γ′}$$

$$\mathbf{Q_{roll-acc}} = 2w_4(\mathbf{A_6}\mathbf{x})\mathbf{A_6} \tag{6.13δ′}$$

Το σύστημα αυτό επιλύεται με αντίστροφη χρονική πορεία (από το τέλος της προσομοίωσης $t = t_{end}$ προς την αρχή $t = t_{start}$) με την **Τερματική Οριακή Συνθήκη** που εξάγεται από τη διαφοροποιημένη εξίσωση της επαυξημένης συνάρτησης κόστους και είναι η εξής:

$$\mathbf{\Psi}^T\big|_{t_{end}} = \mathbf{0} \tag{6.14}$$

Μετά τη λύση των συζυγών εξισώσεων με τη Μέθοδο Runge-Kutta 4ης τάξης, απομένει η εύρεση των **Παραγώγων της Συνάρτησης Κόστους** που δίνονται από τη σχέση:

$$\frac{\delta F}{\delta \mathbf{b}} = \int_{t_{start}}^{t_{end}} \left[\mathbf{\Psi}^T \left(-\frac{\delta A}{\delta \mathbf{b}} \mathbf{x} \right) + \frac{1}{2t_{period}} \mathbf{G}(t) \right] dt - \mathbf{\Psi}^T \frac{\delta \mathbf{x}}{\delta \mathbf{b}}\bigg|_{t_{start}} \tag{6.15}$$

όπου $\mathbf{G}(t)$ είναι το διάνυσμα που περιέχει τους όρους που προκύπτουν από την παρα-

γώγιση των τεσσάρων δεικτών άνεσης:

$$\mathbf{G_{seat-acc}} = 2w_1(\mathbf{A_{16}x} + S_{16})\frac{\delta \mathbf{A_{16}}}{\delta \mathbf{b}}\mathbf{x} \tag{6.16α′}$$

$$\mathbf{G_{seat-jerk}} = 2w_2\left[\mathbf{A_{16}}(\mathbf{Ax} + \mathbf{Bu} + \mathbf{S})\right]\left[\frac{\delta \mathbf{A_{16}}}{\delta \mathbf{b}}\dot{\mathbf{x}} + \mathbf{A_{16}}\frac{\delta \mathbf{A}}{\delta \mathbf{b}}\mathbf{x}\right] \tag{6.16β′}$$

$$\mathbf{G_{pitch-acc}} = 2w_3(\mathbf{A_4x})\frac{\delta \mathbf{A_4}}{\delta \mathbf{b}}\mathbf{x} \tag{6.16γ′}$$

$$\mathbf{G_{roll-acc}} = 2w_4(\mathbf{A_6x})\frac{\delta \mathbf{A_6}}{\delta \mathbf{b}}\mathbf{x} \tag{6.16δ′}$$

Για τον υπολογισμό του τελευταίου αγνώστου όρου $\frac{\delta \mathbf{x}}{\delta \mathbf{b}}\big|_{t_{start}}$ χρησιμοποιείται η παρακάτω γραμμική εξίσωση που λύνεται με τη Μέθοδο της Απαλοιφής Gauss:

$$\mathbf{A}\left.\frac{\delta \mathbf{x}}{\delta \mathbf{b}}\right|_{t_{start}} = -\frac{\delta \mathbf{A}}{\delta \mathbf{b}}\left.\mathbf{x}\right|_{t_{start}} \tag{6.17}$$

Υλοποίηση Αλγορίθμου Βελτιστοποίησης σε Περιβάλλον C++

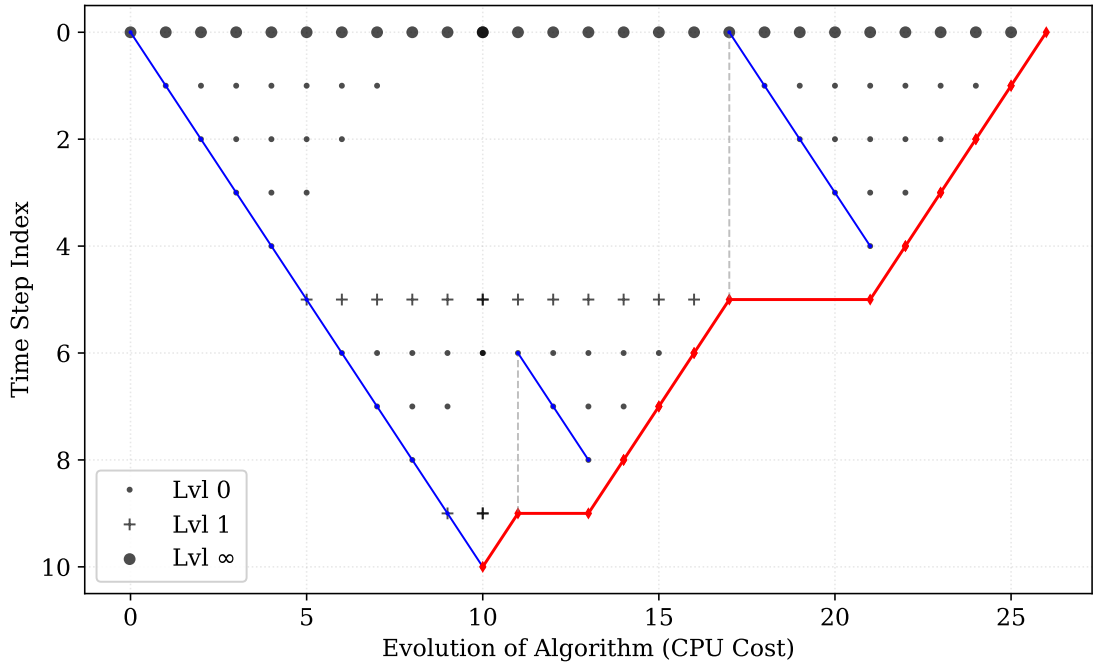
Στην παρούσα εργασία, αναπτύχθηκε ένας εξειδικευμένος επιλυτής σε γλώσσα C++, ο οποίος ενσωματώνει το σύνολο της υπολογιστικής διαδικασίας. Η δομή του είναι η εξής:

- **Primal Solver:** Επιλύει τις 16 διαφορικές εξισώσεις κίνησης του οχήματος στο πεδίο του χρόνου, χρησιμοποιώντας τη μέθοδο Runge-Kutta 4ης τάξης, υπολογίζοντας τις πρωτεύουσες μεταβλητές $\mathbf{x}$.
- **Adjoint Solver:** Επιλύει το σύστημα των συζυγών εξισώσεν με αντίστροφη πορεία, υπολογίζοντας τις συζυγείς μεταβλητές $\mathbf{\Psi}$.
- **Sensitivity Evaluator:** Συνδυάζει τη λύση του πρωτεύοντος και του συζυγούς προβλήματος για τον υπολογισμό των παραγώγων ευαισθησίας $\delta F/\delta \mathbf{b}$.

Οι υπολογισμένες παράγωγοι τροφοδοτούνται στον αλγόριθμο βελτιστοποίησης BFGS (Broyden–Fletcher–Goldfarb–Shanno). Ο BFGS είναι ένας quasi-Newton αλγόριθμος που προσεγγίζει τον πίνακα Hessian των δεύτερων παραγώγων, επιτυγχάνοντας ταχύτερη σύγκλιση σε σχέση με την απλή μέθοδο της απότομης καθόδου (Steepest Descent). Σε κάθε επανάληψη, ο αλγόριθμος προτείνει ένα νέο διάνυσμα μεταβλητών σχεδιασμού, οδηγώντας τη διαδικασία προς το τοπικό ελάχιστο της συνάρτησης κόστους. Η διαδικασία τερματίζεται όταν η μεταβολή της συνάρτησης κόστους ή το μέτρο της κλίσης πέσει κάτω από μια προκαθορισμένη ανοχή (tolerance), υποδεικνύοντας την εύρεση της βέλτιστης λύσης.

Διαχείριση Μνήμης (Check-pointing)

Επειδή η συζυγής επίλυση απαιτεί το ιστορικό της λύσης του πρωτεύοντος προβλήματος (θέσεις και ταχύτητες του οχήματος) σε κάθε χρονική στιγμή, η αποθήκευση όλων των δεδομένων θα απαιτούσε τεράστια μνήμη RAM. Για τον λόγο αυτό, εφαρμόζεται η τεχνική του Check-pointing [17]. Στο Σχήμα 6.3 με μπλε γραμμές διακρίνεται η



Σχήμα 6.3: *Κατανομή των σημείων ελέγχου (checkpoints) για χωρητικότητα μνήμης $S = 5$ σε εύρος 10 χρονικών βημάτων. Κάθε κατακόρυφη στήλη αναπαριστά την κατάσταση της μνήμης σε ένα συγκεκριμένο βήμα του ιστορικού εκτέλεσης του αλγορίθμου.*

επίλυση του Πρωτεύοντος Συστήματος, όπου ο αλγόριθμος ορίζει μια τιμή επιπέδου (level) σε κάθε διάνυσμα μεταβλητών κατάστασης και αποφασίζει με βάση αυτό ποιο σε ποια χρονική στιγμη θα αποθηκεύσει το αντίστοιχο διάνυσμα σε ένα buffer. Όταν η επίλυση του Πρωτεύοντος προβλήματος φθάσει στο τέλος, ξεκινάει η επίλυση του Συζυγούς ανάποδα στον χρόνο και συμβολίζεται με τις κόκκινες γραμμές. Για να βρεθεί η λύση του διανύσματος των συζυγών μεταβλητών $\mathbf{\Psi}$ σε μια χρονική στιγμή t_k, πρέπει και το buffer να διαθέτει το διάνυσμα των πρωτευόντων μεταβλητών $\mathbf{x}$ στην ίδια χρονική στιγμή. Αν δεν το διαθέτει, θα χρησιμοποιήσει το αποθηκευμένο διάνυσμα $\mathbf{x}_{cp}$ που είναι πιο κοντά στη ζητούμενη χρονική στιγμή και θα λύσει εκ νέου το πρωτεύον σύστημα μέχρι εκείνη τη στιγμή. Ο επιλύτης των συζυγών θα το χρησιμοποιήσει, και έπειτα αυτό θα διαγραφεί από το buffer για να πάρουν τη θέση του άλλα διανύσματα $\mathbf{x}$ σε άλλες χρονικές στιγμές. Στο σχήμα φαίνονται όλες αυτές οι διεργασίες, μαζί με τις χρονικές στιγμές που είναι αποθηκευμένες στο buffer σε κάθε βήμα του αλγορίθμου. Τελικά, για ένα buffer με δυνατότητα αποθήκευσης $S = 5$ checkpoints σε συζυγές

πρόβλημα 10 χρονικών βημάτων, θα χρειαστεί να καλέσει 26 φορές έναν επιλύτη για να λύσει το συζυγές και το πρωτεύον πρόβλημα. Στην περίπτωση μας, τα χρονικά βήματα είναι περίπου 10^4 και επιλέχθηκε ένα buffer με δυνατότητα αποθήκευσης 40 στιγμιότυπων. Αυτό έχει ως αποτέλεσμα να χρειαστούν 39291 κλήσεις επίλυσης, και από αυτές οι 19291 οφείλονται στις επανυπολογισμούς των πρωτεύοντος κατά τη συζυγή φάση. Το συνολικό υπολογιστικό κόστος είναι 3,9 φορές μεγαλύτερο από την επίλυση του πρωτεύοντος προβλήματος (από αρχή μέχρι τέλος) μια φορά. Παρ' όλα αυτά, η χρησιμοποιούμενη μνήμη μειώνεται δραστικά κατά 99,6%.

Αποτελέσματα Βελτιστοποίησης

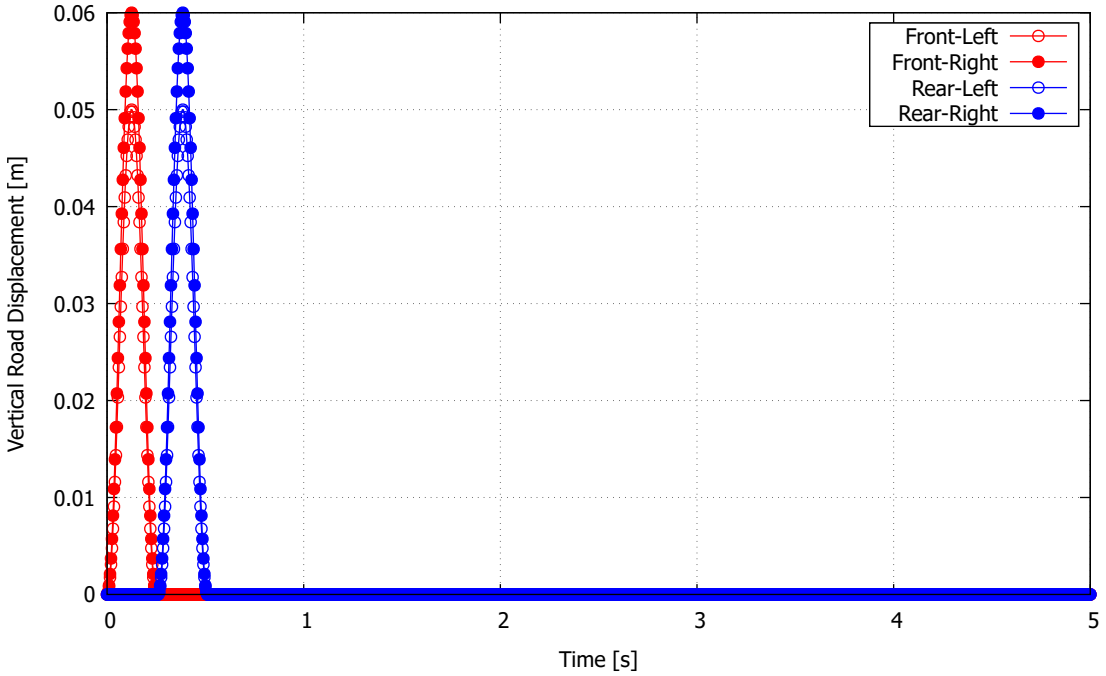
Η βελτιστοποίηση πραγματοποιήθηκε για τρεις διαφορετικές περιπτώσεις διεγέρσεων, αποδεικνύοντας ότι η βέλτιστη ρύθμιση της ανάρτησης εξαρτάται άμεσα από το περιβάλλον οδήγησης.

Περίπτωση Α: Μειωτής Ταχύτητας (Speed Bump)

Στο σενάριο διέλευσης πάνω από ένα μεμονωμένο, ασύμμετρο εμπόδιο (σαμαράκι, όπως φαίνεται στο Σχήμα 6.4), ο αλγόριθμος πέτυχε μείωση της συνάρτησης κόστους κατά 72,2% σε μόλις 15 κύκλους. Ο επιλύτης δημιούργησε ένα όχημα με τις κατάλληλες μεταβλητές σχεδιασμού (Πίνακας 6.2) για να μην αντιστέκεται στο εμπόδιο αλλά να το απορροφάει. Η μέγιστη τιμή της πλευρικής επιτάχυνσης (την οποία προκάλεσαμε με την ασυμμετρία) μειώθηκε κατα 51,00%, το (RMS) της κατά 45,75% και το (Peak) του (jerk) του καθίσματος μειώθηκε κατά 45,99%. Στον Πίνακα 6.3 φαίνονται οι υπόλοιπες μειώσεις για τους δείκτες άνεσης του Σχήματος 6.5.

Πίνακας 6.2: *Σύγκριση Μεταβλητών Σχεδιασμού και Συνάρτησης Κόστους για την Περίπτωση Α*

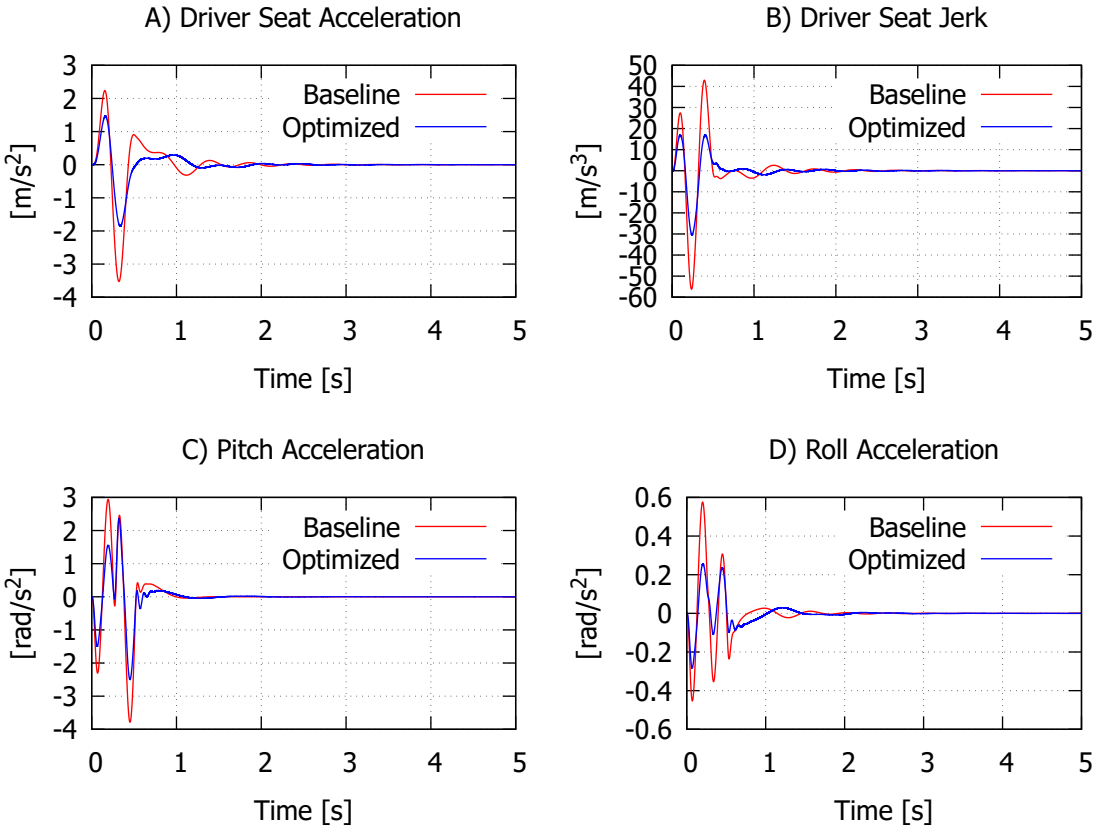
Μεταβλητή	**Αρχική**	**Βελτιστοποιημένη**	**Μεταβολή(%)**
F	2.49629×10^{0}	6.95175×10^{-1}	-72.2%
K_{sf} [N/m]	1.5×10^{4}	7.5×10^{3}	-50.0%
K_{sr} [N/m]	1.7×10^{4}	8.5×10^{3}	-50.0%
B_{sf} [Ns/m]	2.5×10^{3}	1.25×10^{3}	-50.0%
B_{sr} [Ns/m]	2.5×10^{3}	1.25×10^{3}	-50.0%
I_{yy} [kg·m^2]	1.848×10^{3}	1.5×10^{3}	-18.8%
a [m]	1.2	1.04	-13.3%



Σχήμα 6.4: *Οδικό προφίλ μειωτή ταχύτητας στο πεδίο του χρόνου: ασύμμετρο προφίλ (δεξιά-αριστερά),* $v = 10\ m/s$

Πίνακας 6.3: *Ποσοστιαία Μείωση των Δεικτών Άνεσης για την Περίπτωση Α (Μειωτής Ταχύτητας)*

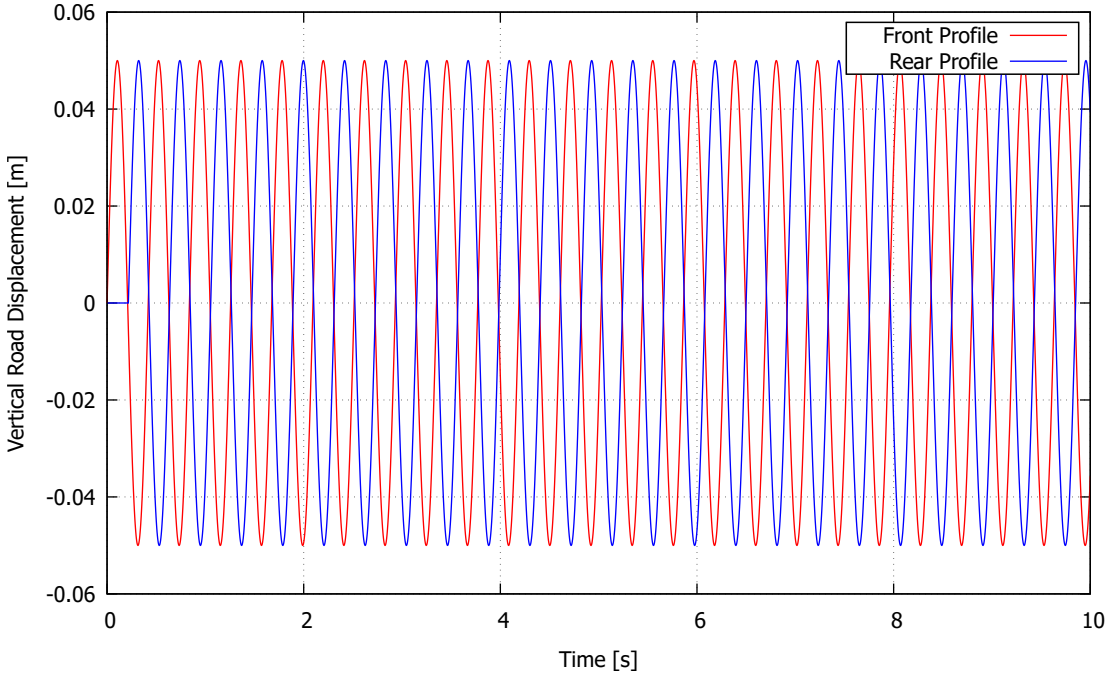
Δείκτης	**Μείωση Peak(%)**	**Μείωση RMS(%)**
Επιτάχυνση Καθίσματος Οδηγού	47.47	42.56
Jerk Καθίσματος Οδηγού	45.99	47.99
Επιτάχυνση Πρόνευσης	34.09	34.09
Επιτάχυνση Πλευρικής Κλίσης	51.00	45.75



Σχήμα 6.5: *Σύγκριση δεικτών άνεσης μεταξύ Αρχικής και Βελτιστοποιημένης διαμόρφωσης για την Περίπτωση Α.*

Περίπτωση Β: Αρμονική-Ημιτονοειδής Διέγερση (Harmonic Road)

Εδώ εξετάστηκε η κίνηση σε οδόστρωμα με συνεχή ημιτονοειδή κυματισμό (Σχήμα 6.6), επιτυγχάνοντας τη μεγαλύτερη βελτίωση με μείωση της συνάρτησης κόστους κατά 89,0% σε 29 κύκλους. Από τον Πίνακας 6.4 φαίνεται ότι ο επιλύτης αύξησε τη ροπή αδράνειας πρόνευσης για να αυξήσει την περιστροφική αντίσταση του αμαξώματος στις συνεχείς ταλαντώσεις, κάνοντας το πιο στιβαρό. Η απόσταση μεταξύ κέντρου βάρους και μπροστά μεριάς του οχήματος a αυξήθηκε το οποίο σημαίνει ότι το Κέντρο Βάρους μετατοπίστηκε προς τα πίσω, τοποθετώντας τον οδηγό κοντά στο μέσο του μεταξονίου, σημείο όπου οι κατακόρυφες μετατοπίσεις λόγω πρόνευσης είναι ελάχιστες. Η RMS τιμή του jerk μειώθηκε εντυπωσιακά κατά 73,08% και της επιτάχυνσης του Καθίσματος του Οδηγού κατά 70,33%. Στον Πίνακα 6.5 φαίνονται οι υπόλοιπες μειώσεις για τους δείκτες άνεσης του Σχήματος 6.7.



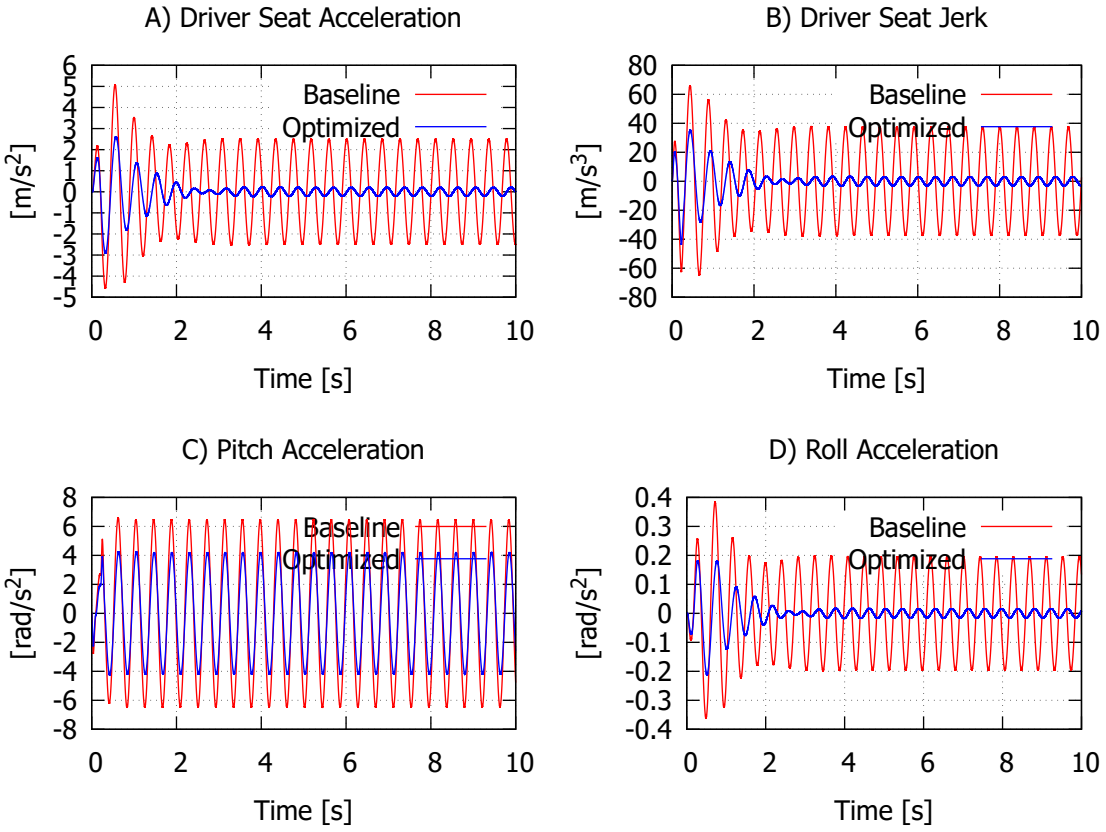
Σχήμα 6.6: *Διέγερση αρμονικού οδικού προφίλ στο πεδίο του χρόνου: συμμετρικό προφίλ (δεξιά-αριστερά), $v = 12\,m/s$*

Πίνακας 6.4: *Σύγκριση Μεταβλητών Σχεδιασμού και Συνάρτησης Κόστους για την Περίπτωση B*

Μεταβλητή	Αρχική	Βελτιστοποιημένη	Μεταβολή(%)
F	2.44198×10^{1}	2.68931×10^{0}	-89.0%
K_{sf} [N/m]	1.5×10^{4}	1.06498×10^{4}	-29.0%
K_{sr} [N/m]	1.7×10^{4}	1.06115×10^{4}	-37.6%
B_{sf} [Ns/m]	2.5×10^{3}	1.61241×10^{3}	-35.5%
B_{sr} [Ns/m]	2.5×10^{3}	2.27492×10^{3}	-9.0%
I_{yy} [kg·m^2]	1.848×10^{3}	2.1×10^{3}	13.6%
a [m]	1.2	1.43	19.2%

Πίνακας 6.5: *Ποσοστιαία Μείωση των Δεικτών Άνεσης για την Περίπτωση B (Αρμονική Διέγερση)*

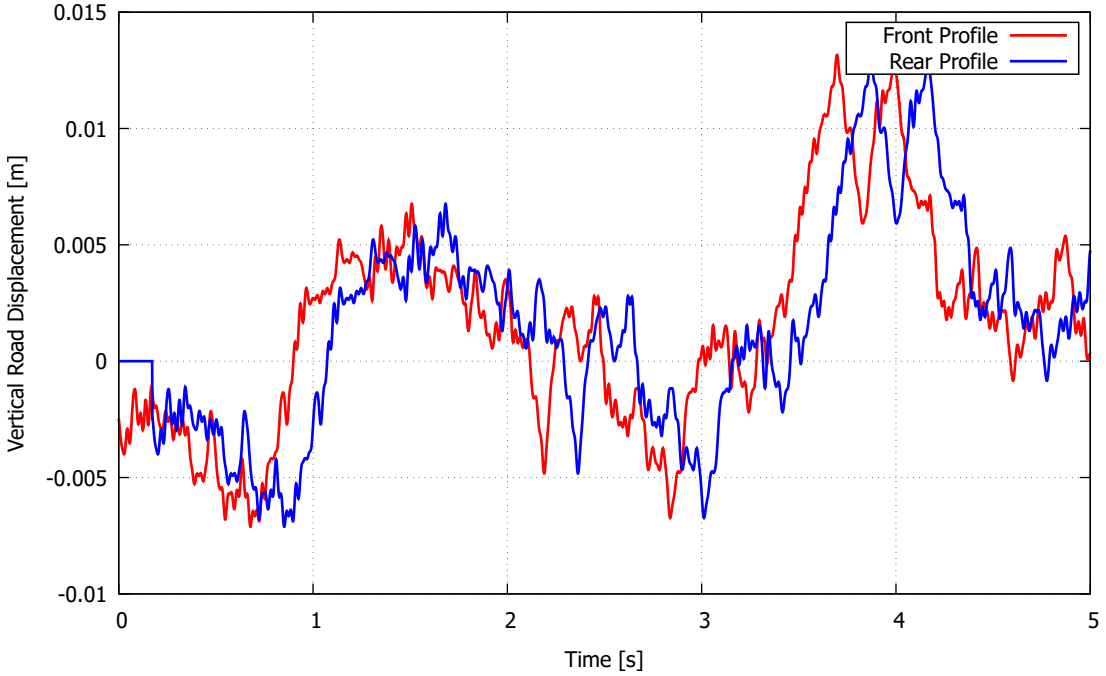
Δείκτης	Μείωση Peak(%)	Μείωση RMS(%)
Επιτάχυνση Καθίσματος Οδηγού	42.38	70.33
Jerk Καθίσματος Οδηγού	34.10	73.08
Επιτάχυνση Πρόνευσης	35.52	34.88
Επιτάχυνση Πλευρικής Κλίσης	44.76	71.12



Σχήμα 6.7: *Σύγκριση δεικτών άνεσης μεταξύ Αρχικής και Βελτιστοποιημένης διαμόρφωσης για την Περίπτωση Β.*

Περίπτωση Γ: Στοχαστικό/Τυχαίο Οδόστρωμα (Stochastic/Random Road)

Στην προσομοίωση τυχαίου οδοστρώματος (Class A) (Σχήμα 6.8), η συνάρτηση κόστους μειώθηκε κατά 62,6% σε 4 κύκλους. Σύμφωνα με τον Πίνακα 6.6, οι αναρτήσεις του μπροστά άξονα ρυθμίστηκαν με χαμηλή ακαμψία και υψηλή απόσβεση για το ομαλή αντιμετώπιση των πρώτων εμποδίων, ενώ οι πίσω αναρτήσεις παρέμειναν πιο σκληρές σε ακαμψία αλλά με χαμηλότερη απόσβεση για τη διατήρηση της ευστάθειας του αμαξώματος. Η RMS τιμή της επιτάχυνσης του καθίσματος μειώθηκε κατά 50,03%, ενώ του jerk κατά 40,90%. Για το σασί, η RMS της πλευρικής κλίσης μειώθηκε κατα 54,85% και της πρόνευσης κατά 9,67%. Στον Πίνακα 6.7 φαίνονται οι υπόλοιπες μειώσεις για τους δείκτες άνεσης του Σχήματος 6.9.



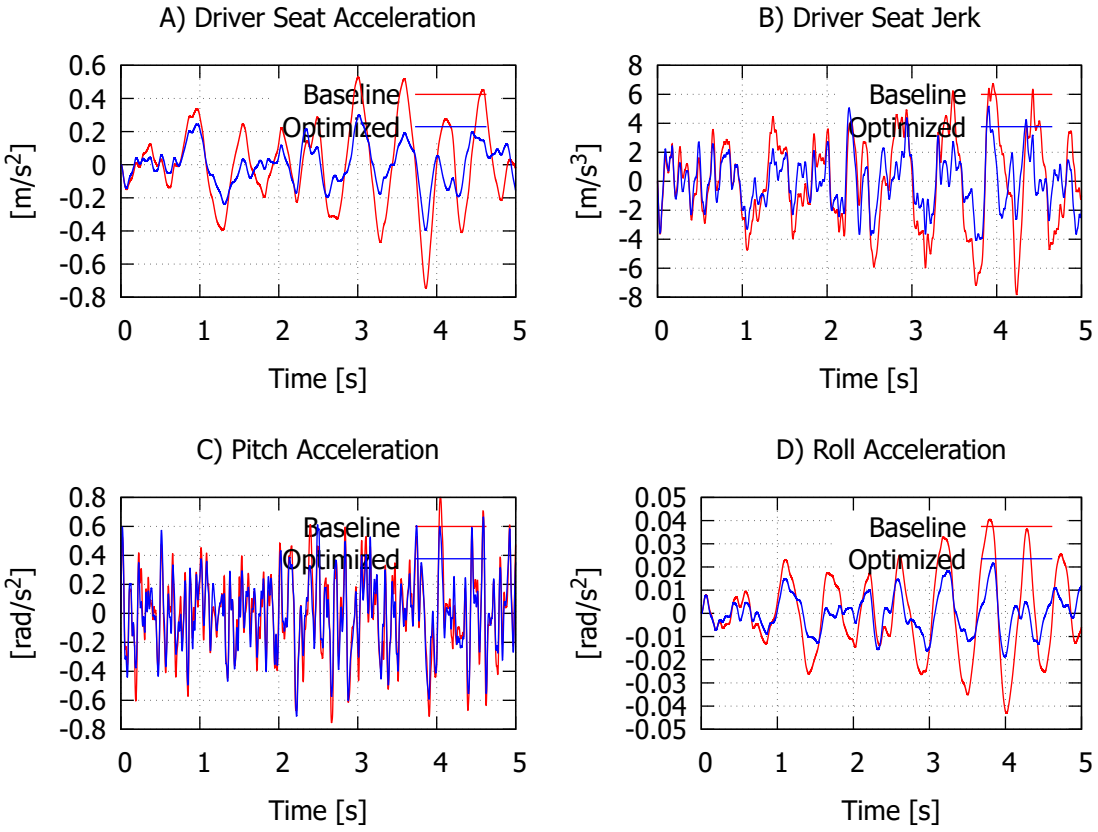
Σχήμα 6.8: *Διέγερση στοχαστικού οδικού προφίλ ISO 8608 Class A στο πεδίο του χρόνου: συμμετρικό προφίλ (δεξιά-αριστερά),* $v = 15\ \mu/\varsigma$

Πίνακας 6.6: *Σύγκριση Μεταβλητών Σχεδιασμού και Συνάρτησης Κόστους για την Περίπτωση Γ*

Μεταβλητή	**Αρχική**	**Βελτιστοποιημένη**	**Μεταβολή(%)**
F	2.9062×10^{-1}	1.08809×10^{-1}	-62.6%
K_{sf} [N/m]	1.5×10^{4}	7.5×10^{3}	-50.0%
K_{sr} [N/m]	1.7×10^{4}	1.68433×10^{4}	-0.9%
B_{sf} [Ns/m]	2.5×10^{3}	2.60928×10^{3}	4.4%
B_{sr} [Ns/m]	2.5×10^{3}	1.62074×10^{3}	-35.2%
I_{yy} [kg·m^2]	1.848×10^{3}	1.92441×10^{3}	4.1%
a [m]	1.2	1.43	19.2%

Πίνακας 6.7: *Ποσοστιαία Μείωση των Δεικτών Άνεσης για την Περίπτωση Γ (Στοχαστική Διέγερση)*

Δείκτης	**Μείωση Peak(%)**	**Μείωση RMS(%)**
Επιτάχυνση Καθίσματος Οδηγού	46.87	50.03
Jerk Καθίσματος Οδηγού	33.95	40.90
Επιτάχυνση Πρόνευσης	11.35	9.67
Επιτάχυνση Πλευρικής Κλίσης	50.01	54.85



Σχήμα 6.9: *Σύγκριση δεικτών άνεσης μεταξύ Αρχικής και Βελτιστοποιημένης διαμόρφωσης για την Περίπτωση Γ.*

Είναι σημαντικό να γίνει αντιληπτό ότι κάθε βελτιστοποιημένη λύση που παρουσιάστηκε εδώ είναι **αποκλειστικά προσαρμοσμένη** στο συγκεκριμένο σενάριο δοκιμής. Στην πραγματικότητα, τα οχήματα καλούνται να αντιμετωπίσουν μεγάλη ποικιλία οδοστρωμάτων και η απόλυτη βελτιστοποίηση για μία μόνο περίπτωση ενδέχεται να οδηγήσει σε ανεπιθύμητη δυναμική συμπεριφορά σε κάποια άλλη. Ο σχεδιασμός ενός οχήματος αποτελεί πάντα έναν συμβιβασμό (trade-off). Συνεπώς, μια επιτυχημένη παθητική ανάρτηση δεν είναι εκείνη που αποδίδει τέλεια σε μία μεμονωμένη κατάσταση, αλλά εκείνη που προσφέρει την καλύτερη δυνατή συμπεριφορά συνολικά (all-around) για το εύρος των συνθηκών λειτουργίας.