

Vehicle Suspension Optimizer: Use of Interactive Evolutionary Computation Using Matlab



Sourav Das^{1*}, Sam Gasper¹, Alex Pallagi¹, Mitchell Schroeder¹, Logan Souders¹ and Caitlin Wolfe²

¹Mechanical Engineering Department, University of Wisconsin-Milwaukee, USA

²Biomedical Engineering Department, University of Wisconsin-Milwaukee, USA

Submission: August 27, 2018; **Published:** October 30, 2018

***Corresponding author:** Sourav Das, (Ph.D. Student), Mechanical Engineering Department, University of Wisconsin-Milwaukee, Milwaukee, USA.

Abstract

Car enthusiasts everywhere value customization and industry performance. A suspension system in a vehicle offers comfort and safety to the passengers. Unevenness in the road condition causing jerk and discomfort to the occupants of the vehicle which directly reflected the steering of the vehicle. The present paper aims for designing a suspension part of the vehicle using MATLAB/Simulink software. Freedom of choice is provided using a simple and easily customizable MATLAB program. This program allows users to select a custom-built suspension system optimized for maximum performance to handle any type of terrain. The program will accept input parameters for car type and terrain and will optimize corresponding spring and shock values. All obtained dampening results are compared with the critically damped value to determine how close the system can be to the ideal value without over or under dampening.

Keywords: Spring; Shock values; Suspension; Optimization and MATLAB- Simulink

Introduction

Automobile suspension is one of the important components used in the vehicle which in general responsible for smooth and controlled driving [1]. Suspension system usually facilitates good friction between the tires and the road and provide stability to the vehicle under static and dynamic conditions and facilitate comfort to the passengers [2]. The suspension system is essentially consisting of a chassis which carries the cab and the springs which absorb impacts resulted due to bad road condition. Hence, without suspension system the impact and shock generated inside the vehicle, due to bad road condition, would be so high that it would be difficult to control the vehicle [3-4].

The suspension system in most cars is designed to handle a specific driving condition with the springs and shocks set by the industry or manufacturer standards. MATLAB / Simulink software was used to simulate and optimize a custom suspension for a specific application condition [5-7]. Design of automobile suspension was carried out using many other design tools [8-12]. Recently, Mitra et al. [12] studied and developed a quarter car simulation model using ADAMS. The parameters used in the modeling were sprung and unsprung mass, tire pressure, speed, spring stiffness, and wheel alignment parameters such as camber and toe. Their study showed that there is a good agreement between ADAMS model and experimental results. Further, the suspension was modeled through half car suspension with 4 degrees of freedom (4-DOF) and the controller used were Proportional Integral Derivatives (PID), Linear Quadratic Regulators (LQR), Fuzzy and Adaptive Neuro-Fuzzy Inference System (ANFIS) using MATLAB-Simulink

[13]. It was reported that ANFIS based controller performed better. Chi et al. [14] reported optimization study of linear quarter car model using three different techniques

- Genetic Algorithm (GA),
- Pattern Search Algorithm (PSA) and
- Sequential Quadratic Program (SQP) using body acceleration, suspension working space, and dynamic tyre load as the designing criterions.

In the present investigation MATLAB/Simulink program was used to analyze the springs and shocks in the vehicle, after applying force and communicate the user about the kind of spring or shock to be used for optimizing a specific system. Usually, the users will be specifying the inputs such as the vehicle's weight, their typical driving conditions, and thereafter one could design and select the optimized spring value. The software program will be facilitating the output of the optimized shock to be used on a vehicle. The user could also provide input on the weight of the vehicle and select the range of optimized shock values. After the program takes in the inputs and outputs, Simulink will graph the displacement and damping behavior of the vehicle's suspension. The system inputs need usually are $f_a(t)$, the force exerted on the wheel by the bump, one-quarter of the car's mass (m), the damping coefficient (B), and the spring constant (K). The program output usually would be in the form of a graph representing the resulting damped created by the shock spring suspension system.

Methods

User inputs are the car types, and different terrain conditions. The program assigned mass (depends on car type) and applied vertical force (depends on terrain). The user then has the option to input, spring choice (K) and/or shock choice (B), or neither and let the system optimize the remaining choices for the best the dampening results based on car mass and terrain. The governing equation of the system was the user-defined function. The user

inputs their car's specifications, and the driving conditions that the car will encounter. The program will require a loop to execute dampened harmonic motion for the given conditions, based on user selections, it will need to optimize the remaining variables selecting the appropriate shock (B) or spring (K) to dampen the system (Figure 1) shows a typical suspension damping diagram three damping conditions i.e., over damping, undamped and critically damped.

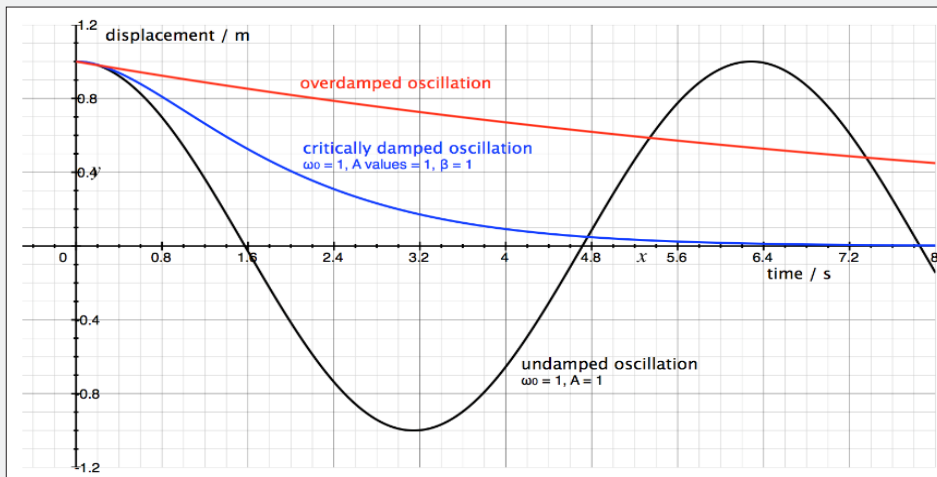


Figure 1: Damping curve shows overdamping, undamped and critically damped conditions.

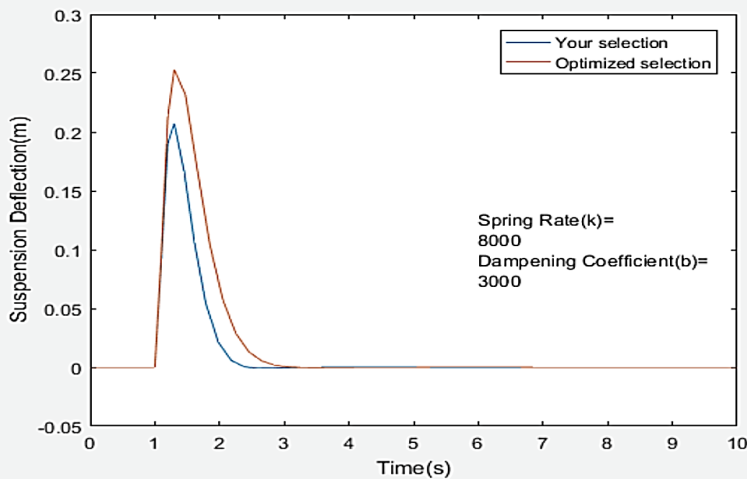


Figure 2a: A generic example of system optimized dampening.

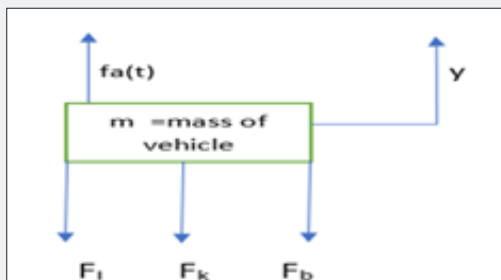


Figure 2b: Modeled damped harmonic oscillator equation.

Four cases were presented in the selection structure of the code that allows the user to pick shock (B) value, pick spring (K) value, pick both, or pick neither. All obtained dampening results were compared to the critically damped value to determine how close the system can be to the ideal value without over or under dampening. After the results were calculated, the values were fed in the program then outputs received as a graph with the suspension deflection over time. If the user, picked both the spring rate and the dampening coefficient then the MATLAB Simulink program will produce output in the form of graphical

representation optimized based on the terrain chosen. An example of the graph is shown below in (Figure 2a & 2b).

$F_i = \text{Inertial Force}$

$$F_i = \frac{md^2y}{dt^2}$$

$F_k = \text{Spring Force Constant}$

$$F_k = K \Delta y = Ky$$

$F_b = \text{Damper Force}$

$$F_b = B \frac{dy}{dt} = By$$

$$my + By + ky = f_a(t)$$

$$y = \frac{1}{m} [fa(t) - By - Ky]$$

The equation set-up was done via Simulink, modeling the forces to be dampened as vertical forces only. The graph output shows a single wheel suspension system, with mass idealized over each wheel as one-quarter of the total vehicle mass (Figure 3). The Under, Over and Critical damping conditions were derived from the characteristic roots, for background on the theory, referenced MIT OCW course number 18.03 [15].

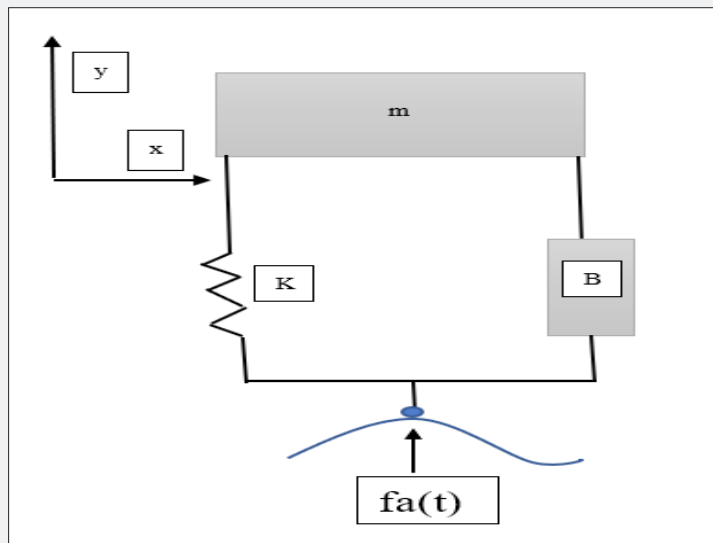


Figure 3: Suspension model for a single wheel with mass over the wheel, idealized as 1/4 of net vehicle mass.

Results & discussion

The program output is shown with two example vehicles, the BMW x Drive35i (Figure 4) and the BMW 440i Gran Coupe (Figure 6). The output shows the user choice program, and the level of dampening, versus the optimized, recommended system choice for critical dampening. The BMW xDrive35i, shown below in

Figure 5 is significantly under dampened, and the user selection is compared to the ideal dampening, recommended by the program. The BMW 440i Gran Coupe, is significantly lighter, and the under dampening effect in similar conditions is not as severe but is apparent in the graph in Figure 7 below. It is also compared to the optimized program selection.

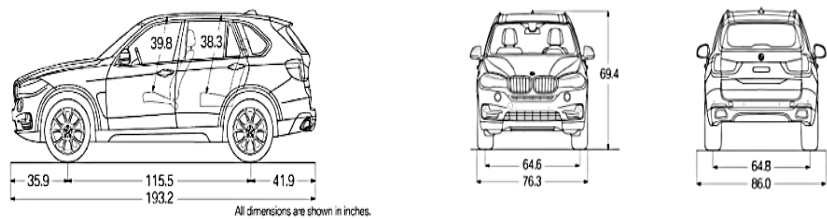
Exterior Dimensions

LENGTH / WIDTH / HEIGHT (in)

193.2 / 76.3 / 69.4

CURB WEIGHT - AUTOMATIC TRANSMISSION (lbs)

4735



X5 xDrive35i mass: 2147.76 kg

Figure 4: BMW xDrive35i mass: 2147.76 kg [16].

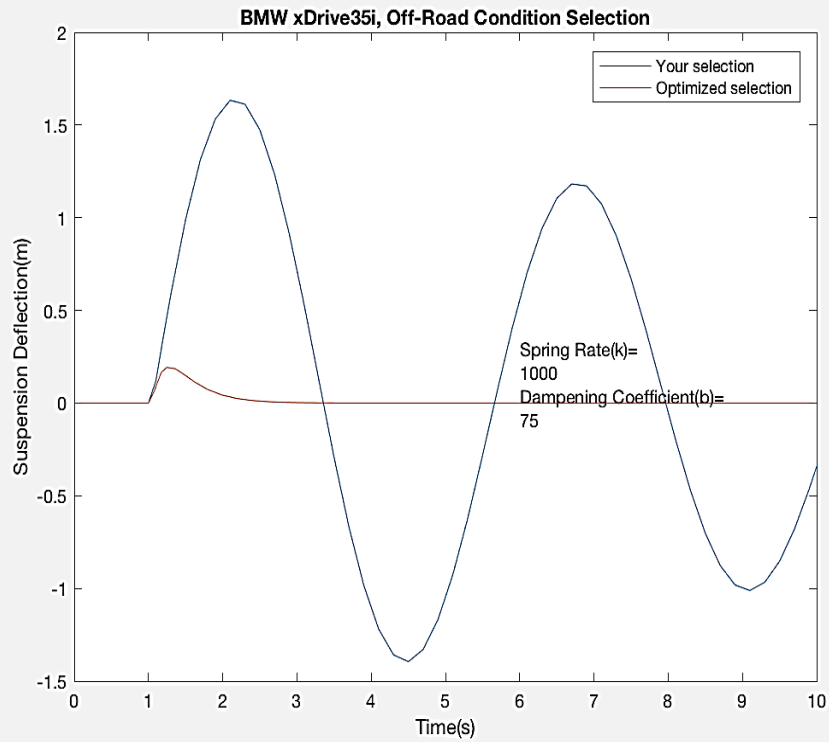
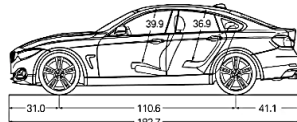


Figure 5: BMW x Drive 35i, off-road condition selection.

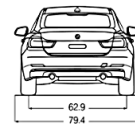
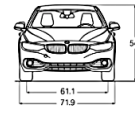
Exterior Dimensions

LENGTH / WIDTH / HEIGHT
(in)
182.7 / 71.9 / 54.7

CURB WEIGHT -
AUTOMATIC
TRANSMISSION (lbs)
3781



All dimensions are shown in inches



440i Gran Coupe
mass: 1715.033 kg

Figure 6: BMW 440i Gran Coupe mass: 1715.033 kg [17].

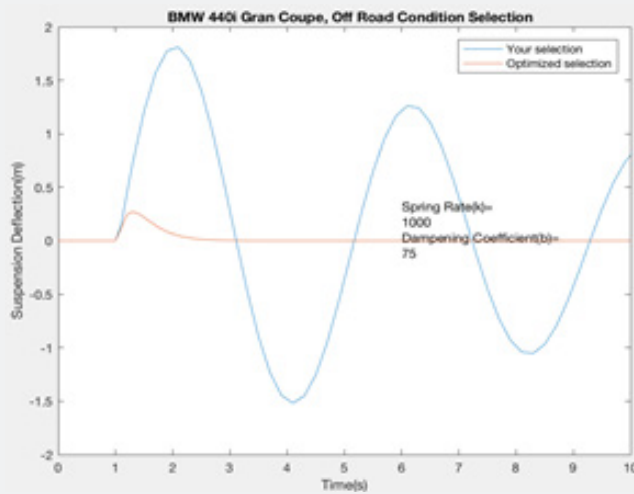


Figure 7: BMW 440i Gran coupe, off-road condition.

Software Structures

Simulink, was used to simulate the vertical motion of the car subject to a bump. The ordinary differential equation modeling the car suspension was graphically represented as a block diagram so that it could be directly entered in Simulink. The block diagram used for the Vehicle Suspension Optimizer is shown in Figure 8. Simulink makes it easy to edit and change equation parameters using the “to, and, from workspace” blocks. The output signal taken from the Simulink model is “y”, the car’s

vertical displacement. Displaying the displacement is a good way to represent the overall performance of the suspension as it encounters a bump. The vertical displacement of a car after hitting a bump is a dampened harmonic oscillation. The plot of displacement shows that after hitting the bump the car will be moving up and then down just one time if the suspension is set up properly. A poor suspension will be oscillating or overshoot meaning the car will travel up and down repeatedly until it eventually stops (Figure 8).

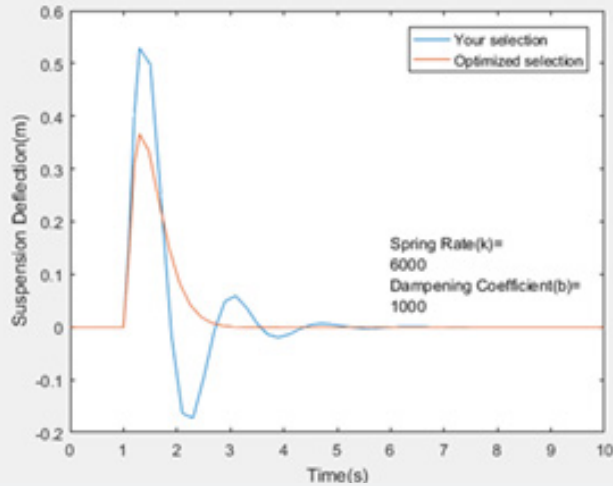


Figure 8: Simulation output.

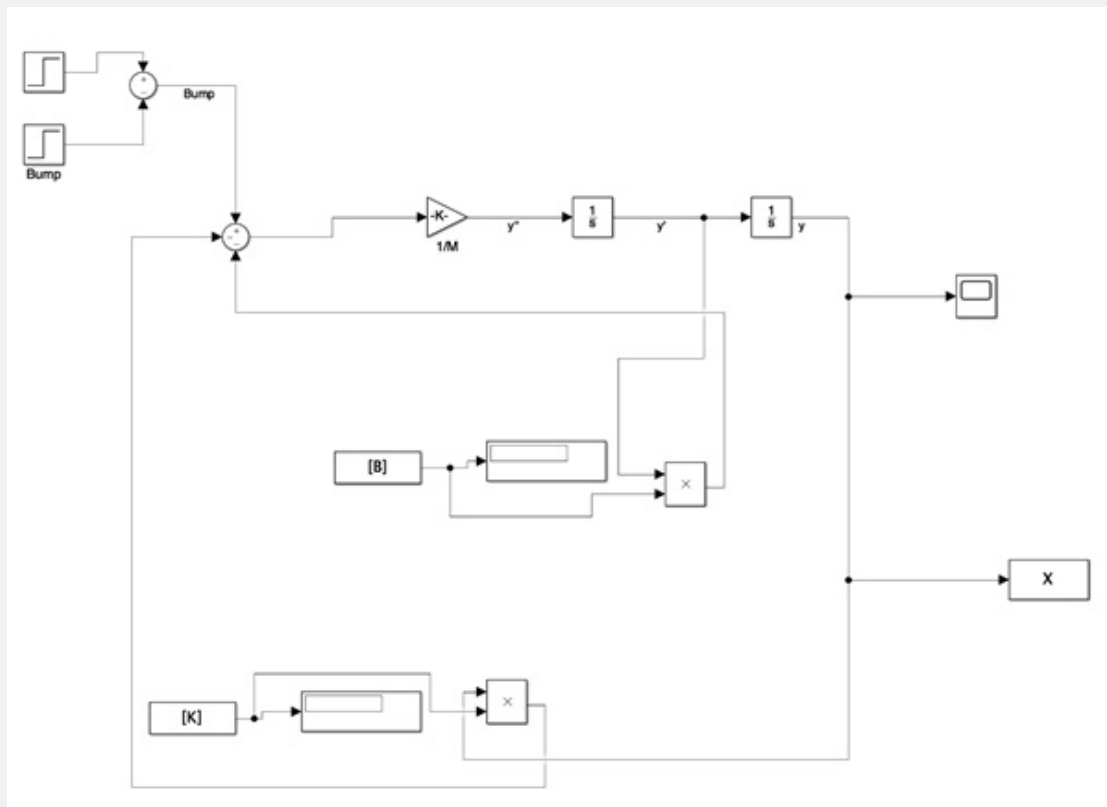


Figure 8: Simulink $My'' + By' + Ky = fa(t)$.

The simulation can be controlled from the MATLAB code directly using the proper commands. The `sim` command was used to start the simulation in the MATLAB code so that the user wouldn't need to press any buttons in the Simulink window. Values

sent to Simulink from the MATLAB workspace must be a vector containing the value plus a time value. Simulink runs a simulation for a specified time, so that the values must include the time that they will be included in the simulation.

GUI Structures

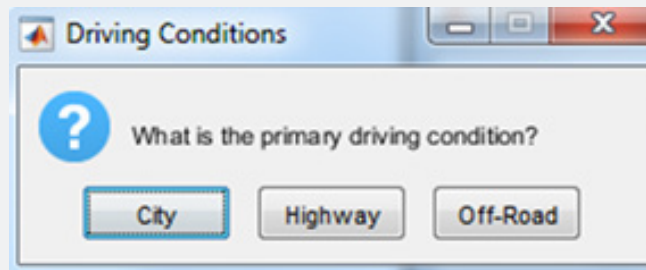


Figure 9: Driving Conditions.

The user interface was handled using integrated MATLAB user dialogue boxes. The first variable, vehicle mass, is generated by the user through interaction with the `inputdlg` command. The command creates a dialogue box with a field for input by the user. The value entered is translated to a double and divided by four to find the vehicle mass at one corner for use in the simulation. The

simulation requires a value of force in Newtons to generate a plot of suspension deflection over a bump. The user is given three options of primary driving conditions via the `questdlg` command, shown in Figure 9 which correlates to force values for the simulated bump. The values for City, Highway and Off-Road driving conditions are 9000, 5000, and 13000 Newtons; respectively.

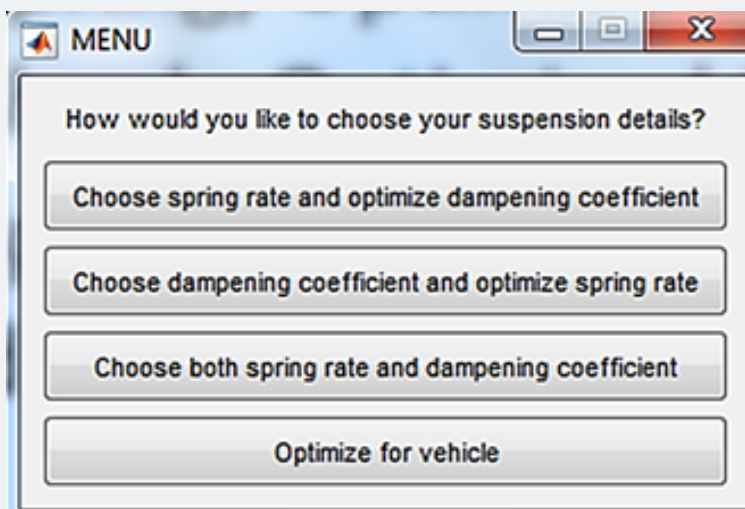


Figure 10: Suspension input details from MATLAB.

The user selection of program function is handled via MATLAB menu command. The user is given four options (as shown in Figure 10) for the simulation to choose based on the component requirements. The program takes known values for spring rate or dampening coefficient and optimizes for the other unknown. The program also uses an `If` statement to determine the optimal spring rate based on the mass of the vehicle, from which it will find the appropriate dampening value to optimize the suspension performance regardless of the user's inputted values. Selection of a known spring rate or dampening coefficient value is handled using the `listdlg` command. Matlab displays a list of available values, which are stored in a string array, from where the user can select one for use in the simulation [Appendix 1]. The suspension deflection over time output from the simulation is displayed via

the native plot function in MATLAB. The user-selected or program-optimized values for spring rate and dampening coefficient are displayed on the plot. If the user selected the option to choose the `K` and `B` values independently, a plot of optimized suspension performance is also displayed with a legend for differentiating the two plots.

Results and Discussion

The aims of this work were to allow user customization, based on driving activities and car types, then to allow the simulated addition of a new spring or shock replacement, or replacement of the suspension altogether. This system was idealized with only one wheel represented in the simulation, with a quarter of the car's mass over the wheel, and a direct vertical shock force.

In more realistic conditions, weight fluctuation based on car orientation and terrain would need to be accounted for, and the fa(t) force may have a horizontal component. Because this system is idealized, and it is a good estimate, but it has some limitations in accuracy for real-world applications. Overall results obtained, in the present study, showed that optimized based dampening, had up to a 10-fold increase in dampening effects, versus direct user choice alone. The program created allowed users to choose from a catalog and customize a suspension to best fit their needs. It also allows for suspension choice based on car mass, and passenger/cargo mass [18]. For example, if the vehicle is class 2b-8a, and must carry unusually heavy or fragile cargo that requires a system more capable of dampening than standard car manufacturer options. This customization may be essential to meet carrier needs. The MATLAB program used in the present study is shown in Appendix.

Conclusion

In the present study automobile suspension was optimized using MATLAB Simulink software. Vehicle mass, and road condition were taken as the input criteria for the optimization study. Damping curve was generated as an output of the simulation. One of the advantages of this simulation method is that it can adapt any kind of model of car and road condition for optimizing the suspension design in the best possible way.

References

1. Automobile suspensions (1981) authored by Colin Campbell, Springer Publication.
2. Sert E, Boyraz P (2017) Optimization of the suspension system and sensitivity analysis for improvement of stability in a midsize heavy vehicle. Eng Sci Tech Int J 20(3): 997-1012.
3. Lai CY, Liao WH (2002) Vibration control of a suspension system via a magnetorheological fluid damper. J Vibration and Control 8(4): 527-547.
4. Werner S, Igor I (2015) Uncertainties in road vehicle suspensions. Procedia IUTAM 13: 151-159.
5. Dhananjay RK (2014) Modeling, Simulation and Control of Semi-Active Suspension System for Automobiles under MATLAB Simulink using PID Controller. IFAC Proceedings Volumes 47(1): 827-831.
6. Anand T, Deshpande AR, Deshmukh SP, Jagtap KR (2014) Modeling, Analysis and PID Controller Implementation on Double Wishbone Suspension Using SimMechanics and Simulink. Procedia Engineering 97: 1274-1281.
7. Gasper S, Pallagi A, Schroeder M, Souders L, et al. MatLab Code Output.
8. Kavitha C, Abinav S, Shankar, B. Ashok, et al. (2018) Adaptive suspension strategy for a double wishbone suspension through camber and toe optimization. Eng Sci Tech an Int J 21(1): 149-158.
9. Thacker S, Bhatt A (2015) Research paper on design and analysis double wishbone suspension system using finite element analysis. Int J Adv Res Eng Sci Tech 2(7): 19-22.
10. Savkin AN, Klimov MA, Doluda AO (2017) The Automobile Design Element Fatigue Life Modeling Due to Its Dynamic Model. Procedia Engineering 206: 416-420.
11. Fengchen T, Quan Y, Caichun H, Lida W (2012) Experimental Study and Design on Automobile Suspension Made of Magneto-Rheological Damper. Energy Procedia 16: 417-425.
12. Anirban CM, Elvis FK, Nawpute SS, Vaibhav K et al. (2018) Development and Validation of a Simulation Model of Automotive Suspension System using MSC-ADAMS. Materials Today Proc 5(2): 4327-4334.
13. Puneet G, Adarsh S, Ramachandran KI (2017) Performance Analysis of Half Car Suspension Model with 4 DOF using PID, LQR, FUZZY and ANFIS Controllers. Procedia Computer Science 115: 2-13.
14. Chi Z, He Y, Naterer G (2008) Design optimization of vehicle suspensions with a quarter-vehicle model. Trans CSME 32(2): 297-312.
15. MIT, "Under, Over and Critical Damping".
16. 2018 BMW USA, "X5 xDrive35i".
17. 2018 BMW USA, "440i Gran Coupe".



This work is licensed under Creative Commons Attribution 4.0 License
DOI: [10.19080/JOJS.2018.01.555574](https://doi.org/10.19080/JOJS.2018.01.555574)

Your next submission with Juniper Publishers will reach you the below assets

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats
(Pdf, E-pub, Full Text, Audio)
- Unceasing customer service

Track the below URL for one-step submission

<https://juniperpublishers.com/online-submission.php>