Final report

ME46060: Engineering Optimisation: Concepts and Applications

Zach Kelly (5901405)
Timotei Dudas (5901421)

Delft University of Technology

TU Delft

Contents

1 | Statement of Contribution

Topic	Involvement Timotei Dudas	Involvement Zach Kelly
Introduction, Background, and Motivation	wrote section	
Modeling Aspects	wrote section	
Optimization Problem	wrote section	
Boundedness	together	together
Monotonicity	together	together
Convexity	together	together
Motivation of Optimization Approach, Choices		wrote section
Investigation of obtained optimum		wrote section
Observations, interpretation of results		wrote section
Conclusions and Recommendations		wrote section

2 | Introduction, Background, and Motivation

Vehicle suspension systems are important to the safe and comfortable operation of a vehicle. A good suspension system often makes the difference between a good and a bad car. This is why there is a large amount of existing research towards improving the safety, reliability, and comfort of these systems.

There are two main types of suspension systems that can be found in most modern-day cars. Passive and active systems that vary in cost and ability. A passive suspension system is one that has set parameters like damping coefficient that are designed into it and do not vary. On the other hand, an active system can be controlled by varying parameters to increase rider comfort. These systems are usually very complex and costly. A passive suspension system can be optimized during the design phase of the vehicle to increase the rider comfort while keeping costs and complexity down. This is the objective of this report.

The two variables that can be optimized are the spring stiffness and damping coefficient of the suspension system. The variation of these values has a large impact on the vertical motion of the vehicle body as it travels over a road surface. A "loose" suspension can result in exaggerated wave-like motion of the vehicle as it travels over bumps which is uncomfortable for passengers. A very "stiff" suspension can result in the force from every bump being directly transferred to the passengers which is jarring and uncomfortable as well. The optimal system is one that balances these two scenarios and is different based on the vehicle type. Race cars need stiff suspension for the best handling performance while regular passenger vehicles require the most comfortable option while maintaining safety.

3 | Problem Formulation

3.1. Modelling aspects

3.1.1. Quarter-car model

The following model is derived from [1].

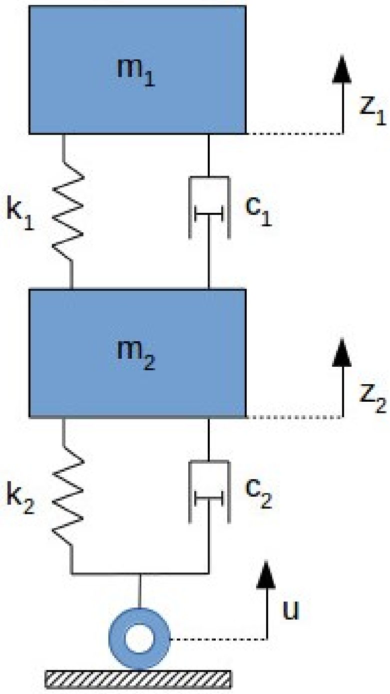

Figure 3.1: Quarter-car lumped parameters model [1]

The system of differential equations that represent the suspension of a quarter-car model is:

$$\begin{aligned} m_1\ddot{z}_1 &= k_1(z_2 - z_1) + c_1(\dot{z}_2 - \dot{z}_1) \\ m_2\ddot{z}_2 &= k_2(u - z_2) + c_2(\dot{u} - \dot{z}_2) - k_1(z_2 - z_1) - c_1(\dot{z}_2 - \dot{z}_1) \end{aligned} \tag{3.1}$$

where:
m_1 [kg] - the mass of a quarter of the car body
m_2 [kg] - the mass of the wheel and suspension
k_1 [N/m] - spring constant (stiffness) of the suspension system
c_1 [N·s/m] - damping constant of the suspension system
k_2 [N/m] - spring constant (stiffness) of the wheel and tire
c_2 [N·s/m] - damping constant of the wheel and tire
z_1 [m] - displacement of the vehicle body (output)
z_2 [m] - displacement of the wheel (output)
u [m] - road profile change (input))

Laplace transform is applied to Equation 3.1 with the initial conditions set to zero, resulting in:

$$Z_1(s)\ (m_1 s^2 + c_1 s + k_1) \;=\; Z_2(s)\ (c_1 s + k_1) \tag{3.2}$$

$$Z_2(s)\ (m_2 s^2 + k_2 + c_2 s + c_1 s) \;=\; U(s)\ (c_2 s + k_2) + Z_1(s)\ (k_1 + c_1 s) \tag{3.3}$$

From Equation 3.1 we have a system of equations that can be used to find a transfer function for our desired output z_1, which is the displacement of the quarter-car mass.

We can solve Equation 3.2 for $Z_2(s)$:

$$Z_2(s) \;=\; Z_1(s)\ \frac{m_1 s^2 + c_1 s + k_1}{c_1 s + k_1} \tag{3.4}$$

Substituting Equation 3.4 into Equation 3.3 results in the transfer function:

$$H_1(s) \;=\; \frac{Z_1(s)}{U(s)} \;=\; \frac{c_1 c_2 s^2 + (k_1 c_2 + k_2 c_1) s + k_1 k_2}{m_1 m_2 s^4 + (m_1 c_1 + m_1 c_2 + m_2 c_1) s^3 + (m_1 k_1 + m_1 k_2 + k_1 m_2 + c_1 c_2) s^2 + (c_1 k_2 + k_1 c_2) s + k_1 k_2} \tag{3.5}$$

where the displacement of the quarter-car body mass, $z_1(t)$ is the output and the randomly generated road profile, $u(t)$ is the input.

In Table 3.1 below the model parameter values are shown. These values are taken from [2].

Table 3.1: Quarter-car model parameters

Parameter	Value
m_1	500 kg
m_2	30 kg
k_1	20,000 N/m
c_1	1,500 N·s/m
k_2	200,000 N/m
c_2	1 N·s/m

The transfer function model is realized using the Matlab function $lsim$ and $step$ which outputs the simulated response of Equation 3.5 to a given set of time samples and input signal. Figure 3.2 below shows the response of the system to a step signal input representing the displacement of the quarter-car mass as a result of driving over a 0.1 m edge.

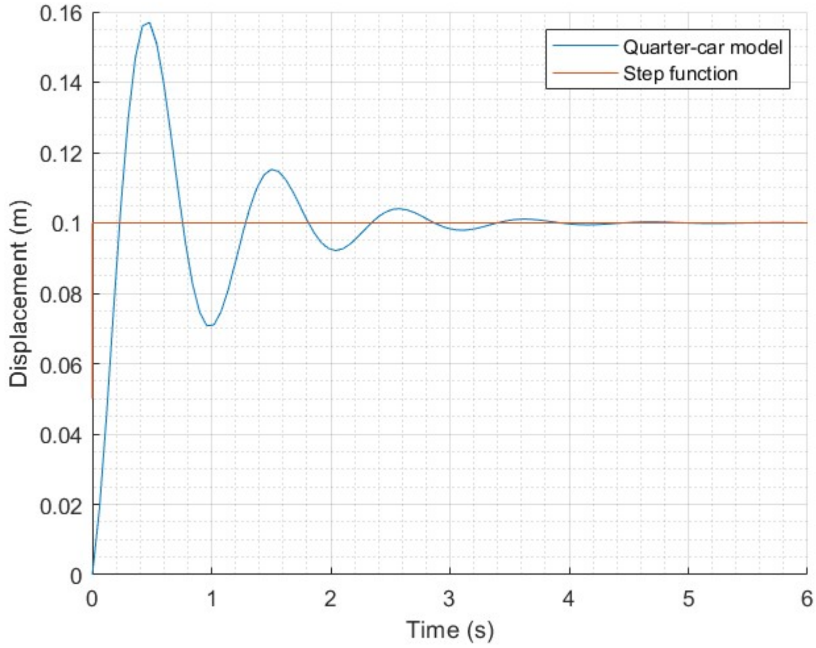

Figure 3.2: Displacement step response of quarter-car model

The desired output of the system is the acceleration of the quarter-car body mass and not the displacement. The acceleration can be found by taking the second derivative of the displacement with respect to time. This is done with the Matlab function $diff$ which outputs the difference between the elements in the displacement array. This is then divided by the time interval between data points. After completing this calculation twice, the acceleration of the quarter-car mass is found.

The step response of the model after differentiation once to achieve velocity is shown in Figure 3.3 below. After a second differentiation, the step response of the acceleration of the quarter-car mass is shown in Figure 3.4.

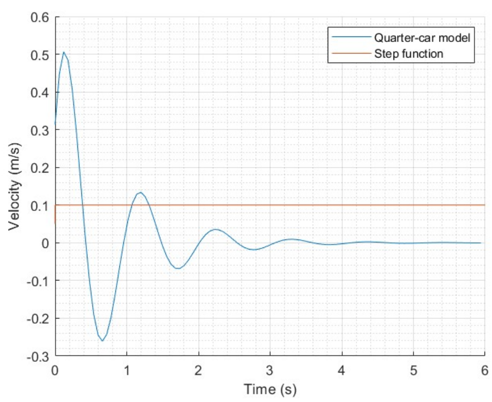

Figure 3.3: Velocity step response of the quarter-car model

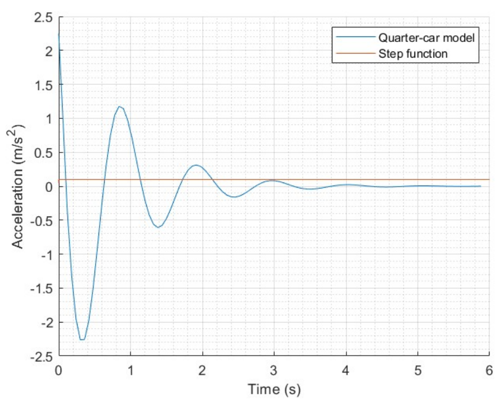

Figure 3.4: Acceleration step response of the quarter-car model

Now that the quarter-car suspension model is complete, a realistic input to the model needs to be defined. This is described in the next section.

3.1.2. Random road profile

Although the step response of the quarter-car model allows for some analysis and visualization of the effect of spring constant and damping coefficient on the acceleration, a more realistic input is beneficial. A randomly generated road profile allows for simulation over more realistic conditions and more optimization potential.

A random road profile based on power spectral density is shown in Figure 3.5. This profile is achieved using the Matlab script in [3]. The mathematical derivation used to achieve the random road profile is outside of the scope of this report. However, the parameters that are important to this application and can be varied include the number of samples, N, the speed of the car, v, the distance traveled, d, and $G_d(\Omega_0)$ which corresponds to the class of the road determined by ISO 8608 [4].

Table 3.2: Random road profile parameters

Parameter	Value
N	1000
v	80 km/h
d	250 m
$G_d(\Omega_0)$	$2x10^{-6}$ m^3 (Class A)

Using the parameters in Table 3.2, a random road profile is made. This road profile is based on ISO 8608 road class A which means that it represents the best quality road with the least bumps and waves. In order to use this road profile as an input to the quarter-car model, the displacement of the road must be relative to time and not distance. This is easily achieved by dividing the traveled distance by the speed.

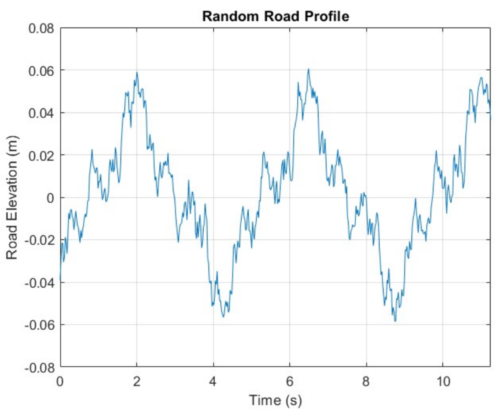

Figure 3.5: Random road profile over time at 80 km/h

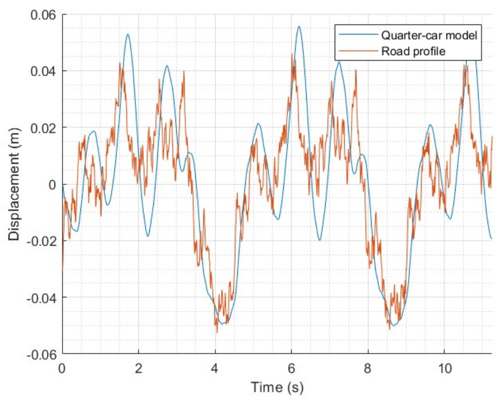

Figure 3.6: Displacement response to random road profile

The output of the quarter-car model before differentiation when using this road profile as the input can be seen in Figure 3.6. This shows the displacement of the quarter-car mass as it travels over the road surface. After the first differentiation, Figure 3.7 shows the vertical velocity of the quarter-car mass. The acceleration output that will be used in the optimization process is shown in Figure 3.8. The acceleration response shown in this figure is specific to the initial conditions of spring constant and damping coefficient that were set in the previous section. Altering the parameters of the Matlab script and simulating over different values of speed and road class will allow for optimization and comparison.

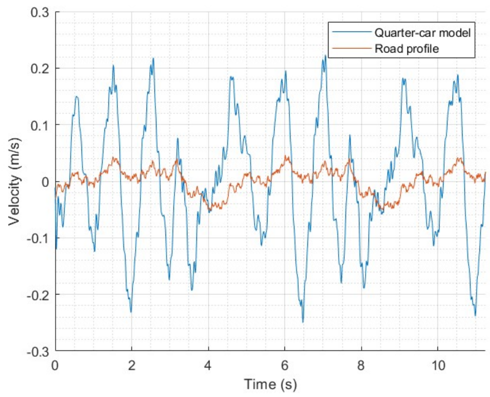

Figure 3.7: Velocity response to random road profile

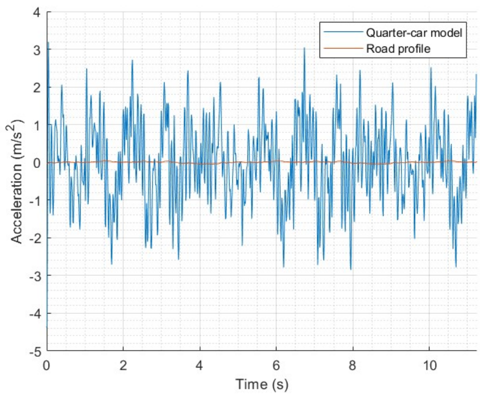

Figure 3.8: Acceleration response to random road profile

3.2. Optimization problem

Now that the the quarter-car suspension system has been modelled and the random road profile input signal has been defined, an optimization problem can be set up.

When optimizing a suspension system, there are many possible goals and outcomes but one of the most important indicators as to whether or not a suspension system is good is how much acceleration (force) that the passengers experience. Minimizing this acceleration as a car travels over a road surface can be accomplished in different ways. The root mean square or the maximum acceleration value can be used. Using the root mean square has the advantage of minimizing the overall acceleration experienced over a given period of time but this method can allow for peaks in acceleration. These peaks affect passenger comfort much more than the smaller vibrations and bumps. This is why the maximum acceleration values will be used for optimization.

The optimization problem based on the model developed in the section above is shown mathematically in Equation 3.6 below.

$$\begin{aligned} \min_{k_1, c_1} \quad & \ddot{z}_1 = f(z_1, z_2, \dot{z}_1, \dot{z}_2, k_1, k_2, c_1, c_2) \\ \text{subject to} \quad & k_{1min} \leq k_1 \leq k_{1max}, \\ & c_{1min} \leq c_1 \leq c_{1max}. \end{aligned} \tag{3.6}$$

The design variables in this problem are k_1 and c_1 which represent the spring constant (stiffness) and damping constant of the suspension system, respectively. The other inputs to the system are considered parameters and are shown in Table 3.1 above.

The problem is constrained by the ranges of possible k_1 and c_1 values. In this report, 9000 N/m to 30,000 N/m will be used for k_1 and 980 N·s/m to 4300 N·s/m will be used for c_1 [5]. These constraints are needed to prevent unrealistic optimization results such as having a spring or damping constant of zero or very large values that do not exist in real suspension systems.

4 | Initial Problem Investigation

This section investigates the objective function Equation 3.6 in order to verify that the optimization problem is formulated well, to explore possible simplifications, and to provide direction when choosing suitable optimization algorithms in Chapter 5.

4.1. Boundedness

A bounded objective function is necessary to avoid unrealistic solutions. To test the boundedness of Equation 3.6, the following sets are defined:

$$N = \{x : 0 \leq x \leq \infty\}$$
$$P = \{x : 0 < x < \infty\}$$

where N is the set of non-negative numbers and P is the set of positive and finite numbers.

If the greatest lower bound for non-negative inputs has a minimizer that exists, is positive, and finite, the objective function is well-bounded. In Figure 4.1, the greatest lower bound and the minimizer associated with it are shown.

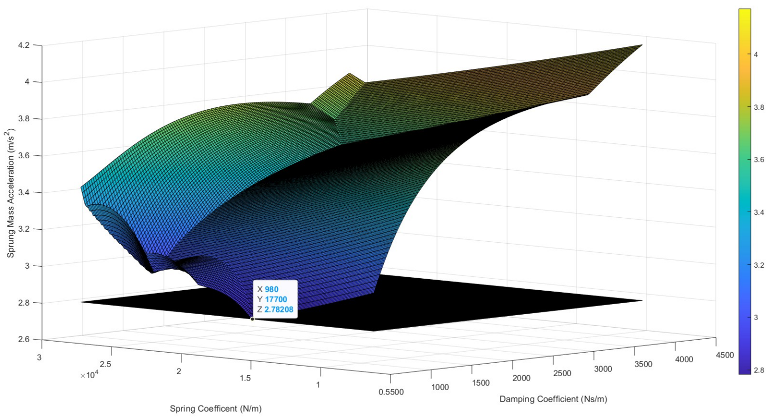

Figure 4.1: Greatest lower bound and minimizer

$$\begin{aligned}
\bar{l}^{+} &= 2.782 \quad k_1, c_1 \in N \\
C_1 &= \{c_1 : f(c_1) = \bar{l}^{+}\} \; = \; 980 \\
K_1 &= \{k_1 : f(k_1) = \bar{l}^{+}\} \; = \; 17{,}700 \\
\varnothing &\neq C_1, K_1 \; \subseteq \; P
\end{aligned} \tag{4.1}$$

Equation 4.1 above shows the proof that the objective function Equation 3.6 is well-bounded. This means that no additional constraints are needed to prevent unrealistic solutions.

4.2. Monotonicity

The objective function Equation 3.6 is monotonic if it solely increases or solely decreases with k_1 and c_1. For a minimization problem like this to be well-constrained, every variable that increases the function must be bounded by at least one non-increasing active constraint.

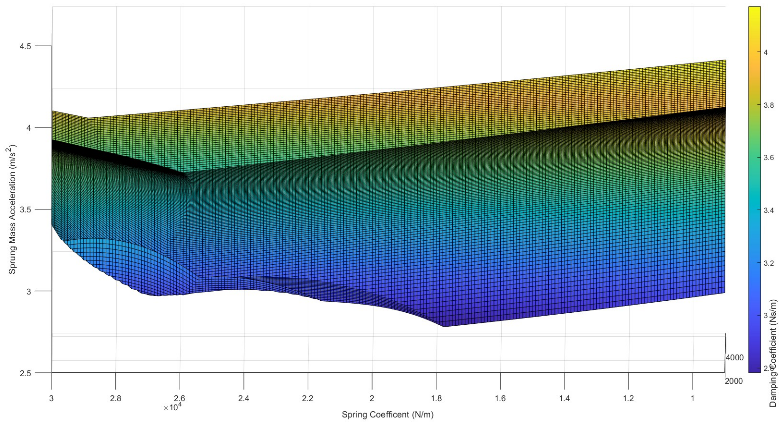

Figure 4.2: k_1 side-view of the objective function

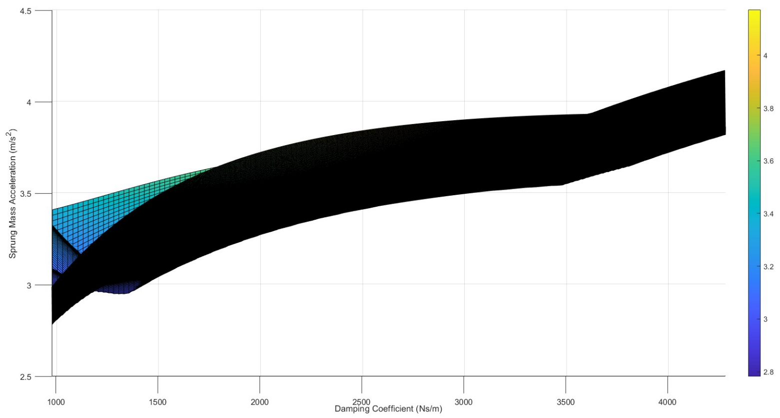

Figure 4.3: c_1 side-view of the objective function

In Figure 4.2 and Figure 4.3, it can be seen that the function does not solely increase or solely decrease with k_1 and c_1. This means that the objective function is non-monotonic and therefore, well-constrained. There are no critical constraints that need to be eliminated or modified.

4.3. Convexity

The objective function is not convex as can be observed in both Figure 4.2 and Figure 4.3. There are multiple points on the function that cannot be connected by a line without crossing underneath a different part of the function. This means that there are multiple local boundary minimums that an optimization algorithm could get stuck in. This is useful information to know when choosing the optimization method in Chapter 5 below.

5 | Optimization

5.1. Motivation of optimization approach, choices

5.1.1. Brute force method

The brute force method is relatively simple to implement and guarantees global optimum within the domain. Considering that our objective function is non-convex and contains local minima, the guaranteed global optimum is held to be an important property. The results obtained from this method will provide a good benchmark for assessing other algorithmic solutions and performance.

5.1.2. Nelder-Mead simplex method

The Nelder-Mead simplex method has an improved convergence rate over other direct search methods such as the brute force method and does not make assumptions about the smoothness, convexity or differentiability of the objective function. The method does not require derivative information, therefore approximation errors resulting from the numerical differentiation of the objective function can be avoided.

5.1.3. Cross search method

The cross search method has improved convergence rate over the brute force method when the parameters are chosen well and does not depend on convexity or differentiability of the objective function.

5.2. Investigation of obtained optimum

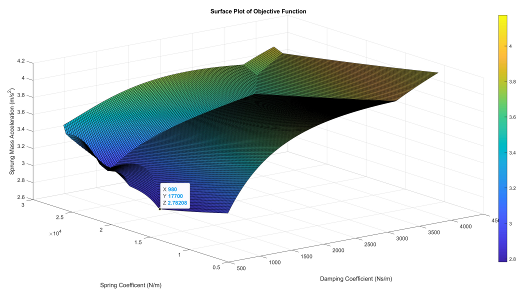

Figure 5.1: Surface plot of global optimum at 80 km/h

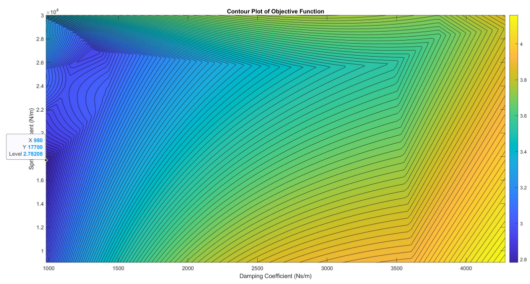

Figure 5.2: Contour plot of global optimum at 80 km/h

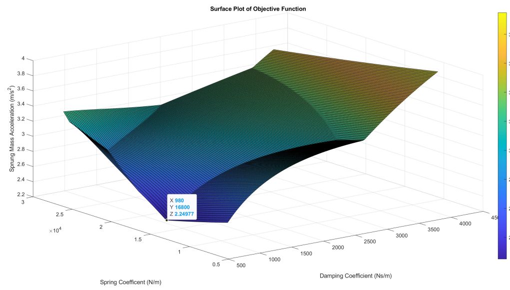

Figure 5.3: Surface plot of global optimum at 60 km/h

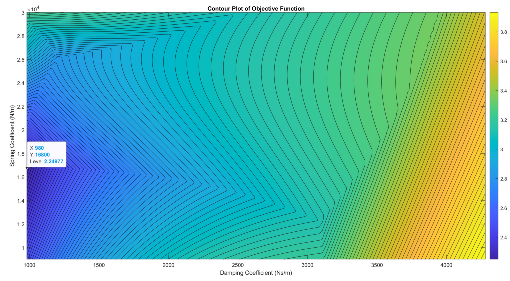

Figure 5.4: Contour plot of global optimum at 60 km/h

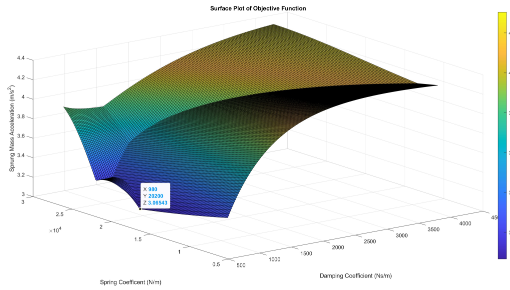

Figure 5.5: Surface plot of global optimum at 100 km/h

Figure 5.6: Contour plot of global optimum at 100 km/h

5.3. Observations, interpretation of results, conclusions

In Table 5.1, the results of the different algorithms are shown.

	Nelder-Mead Simplex	Brute Force	Cross Search
Spring Stiffness (N/m)	980	980	1000
Damping Coefficent ($N{\cdot}s/m$)	17767	17700	9800
Acceleration (m/s^2)	2.7807	2.7821	2.9953
Computational Time (s)	0.330	37.3	2.71

Table 5.1: Algorithmic results and performance

The Nelder-Mead method displayed significant improvement over the brute force and cross search method in terms of computational time as it converged in 0.33 seconds compared to 37.3 and 2.71 seconds. This attributes to the fact that the brute force method is constraint to only global changes in search space whereas the nelder-mead method is able to change its search space locally and have less impact on the convergence rate. The nelder-mead method is superior to the cross search method in that the simplex shape changes size in response to local changes, whereas the cross shape changes size regardless of the local conditions. Due to this property, the cross search method convergence depends on the defined parameters and initial conditions.

6 | Conclusions and Recommendations

According to the results, the most optimal design parameters for spring stiffness and damping resistance if the vehicle is travelling at 60 km/h and subject to a sinusoidal road profile are 980 N/m and 20,200 N·s/m. The maximum acceleration experienced by the car at those conditions is 2.25m/s^2.

The resulting optimums were obtained via zeroth order methods. Higher order methods were not chosen as they require additional numerical differentiation leading to more numerical approximation error. More accurate solutions can be obtained with zeroth order methods at little computational expense, hence the low dimensionality of the objective function.

Modelling the systems acceleration as an analytical expression would be advantageous as it will reduces errors from numerical differentiation and simplify the implementation of optimization algorithms.

link to repository: https://github.com/Zach-K408/spring_damper_optimization

Bibliography

[1] Quarter car suspension model transfer function. *x-engineer.org*. URL: https://x-engineer.org/quarter-car-suspension-transfer-function/ (visited on 06/24/2023).

[2] V. Paliwal, R. Dobriyal, N. Sengar, and P. Kumar. “Optimization of the Suspension Parameters of a Quarter Car Model with Nonlinear Damper”. In: *Advances in Engineering Design* (2021), pp. 489–495.

[3] M. Sabbadini. *RandomRoadProfile, GitHub Repository*. 2023. URL: https://github.com/MaximeSabbadini/RandomRoadProfile/tree/main (visited on 06/18/2023).

[4] M. Agostinacchio, D. Ciampa, and S. Olita. “The vibrations induced by surface irregularities in road pavements a Matlabő approach”. In: *European Transport Research Review* Vol. 6 (2013), pp. 267–275.

[5] D. Koulocheris and C. Vossou. “A Comparative Study of Integrated Vehicle-Seat-Human Models for the Evaluation of Ride Comfort”. In: *Vehicles* Vol. 5 (2023), pp. 156–176.