Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 150319, 10 pages
http://dx.doi.org/10.1155/2014/150319

Research Article

Firefly Optimization and Mathematical Modeling of a Vehicle Crash Test Based on Single-Mass

Andreas Klausen,[1] Sondre Sanden Tørdal,[1] Hamid Reza Karimi,[1] Kjell G. Robbersmyr,[1] Mladen Jecmenica,[2] and Ole Melteig[2]

[1] *Department of Engineering, Faculty of Engineering of Science, University of Agder, 4898 Grimstad, Norway*
[2] *Department of Technology, Institute for Process Technology, Telemark University College, 3914 Porsgrunn, Norway*

Correspondence should be addressed to Hamid Reza Karimi; hamid.r.karimi@uia.no

Received 7 May 2014; Accepted 9 June 2014; Published 22 June 2014

Academic Editor: Weichao Sun

Copyright © 2014 Andreas Klausen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper mathematical modeling of a vehicle crash test based on a single-mass is studied. The model under consideration consists of a single-mass coupled with a spring and/or a damper. The parameters for the spring and damper are obtained by analyzing the measured acceleration in the center of gravity of the vehicle during a crash. A model with a nonlinear spring and damper is also proposed and the parameters will be optimized with different damper and spring characteristics and optimization algorithms. The optimization algorithms used are interior-point and firefly algorithm. The objective of this paper is to compare different methods used to establish a simple model of a car crash and validate the results against real crash data.

1. Introduction

Crash worthiness is the study of the response of a vehicle during an impact on another car, an obstacle, or a pedestrian. In most of the cases, the damage on the cars' occupants is of most importance when studying a vehicle crash. When the car manufacturers are facing the end phase of designing a vehicle, several prototypes of the vehicle needs to be built and crashed in order to monitor the effects of the occupant in different crash scenarios. This process requires a lot of time, production cost, and trained personnel to perform the crash and analyze the data. Because of this, it is more beneficial to perform a couple of crashes and develop a mathematical model of the car to simulate crashes instead.

In order to make a model of the vehicle, there are generally two categories; finite element method (FEM) models and lumped parameter models (LPM). FEM models of the vehicle are created in various CAD software and contain detailed material responses during a crash, but they are generally harder to model correctly, and time simulations are often time consuming, although the results may seem to be satisfactory. Some recent studies in crashworthiness use FEM models to establish the model. Sun et al. [1] studied thin walled structures and energy absorption with different wall thicknesses across its length, which is crucial for occupant safety. Peng et al. [2] looked at different windshield models to predict an impact of a pedestrian's head compared to experimental data. Al-Thairy and Wang [3] developed a simplified analytical model to predict the critical velocity of transverse impact on steel columns. Another study that looks at maximizing the crash energy absorption has been done by Kim et al. [4] which optimizes the design of the motor room in order to maximize energy absorption. Liao et al. [5] also optimized a vehicle design to increase the energy absorption of the vehicle during frontal and oblique crashes. These studies show why designers should keep energy absorption in a vehicle crash of high importance to keep pedestrians and occupants safe.

Unlike a FEM model, a lumped parameter model utilizes several masses in connection with springs and dampers to simulate the response. Together with experimental crash data, parameter values for the springs and dampers can be determined. Recently, several different lumped parameter models were developed in the literature. A Maxwell model

was also presented in [6] which showed great results. A Maxwell model is also well suited to simulate component relaxation and creep [7]. They also presented a model based on an elastoplastic spring [8] which has nonlinear characteristics depending on the status of loading or unloading the spring. In a more recent study of a vehicle LPM, Elmarakbi et al. [9] proposed a mathematical model of two vehicles in a front collision with or without active breaking systems and active suspensions systems.

It is worth noting that the parameters for lumped parameter models can be found by using optimization algorithms. There are many optimization algorithms that can be used for parameter identification. Yang and He [10] proposed a nature inspired metaheuristic algorithm in 2008 that is based on attraction between fireflies. This algorithm was further improved upon when Gou and Wang [11] proposed a hybrid approach between the firefly algorithm and harmony search for global numerical optimization problems. Tang et al. [12] proposed a novel random search algorithm for discrete variables which they used to successfully optimize a vehicle frontal member.

LMP and FEM models are just a couple of ways to reproduce real crash test data. Other methods rely on training a neural network into reproducing kinematics of the vehicle. In [13] they trained an adaptive neurofuzzy inference system (ANFIS) based approach to reproduce kinematics from real crash data. This was done for a 3-degree-of-freedom (DOF) oblique crash and a single-DOF frontal crash. In [14] they trained a neural network to estimate parameters which could reproduce real crash data under different initial velocities. In [15] they used a Wavelet transform to model a real vehicle crash with good precision. In [16] they used a Levenberg-Marquardt algorithm to estimate parameters for a 2-DOF Maxwell model. In [17] they presented three different regressive models, namely, RARMAX, ARMAX, and AR, which were used to estimate physical parameters and reproduce car kinematics.

In this paper we present five different mathematical models whose parameters are estimated by using experimental crash data. The main purpose of this paper is to compare simulations versus crash data and determine which model fits the crash data best. The mathematical models are not derived from analyzing the car body but merely attempts in interpreting the physical response into simple models. Section 2 describes the experimental crash data in use, while in Section 3 we present the different mathematical models. Section 5 shows simulation results and comparisons with real crash data and a conclusion is made in Section 6.

2. Experimental Crash Test Data

The experimental crash test data that is used for parameter identification of the models is collected from a calibration test on a Ford Fiesta that crashed in a pole [18]. The car was equipped with an accelerometer in its center of gravity that measured the acceleration signals in the x, y, and z directions. The acceleration in the x direction is used to identify the parameters for the models in this paper. In order to find the velocity and displacement of the vehicle, a Forward-Euler time integration scheme is applied to the acceleration data with these initial conditions:

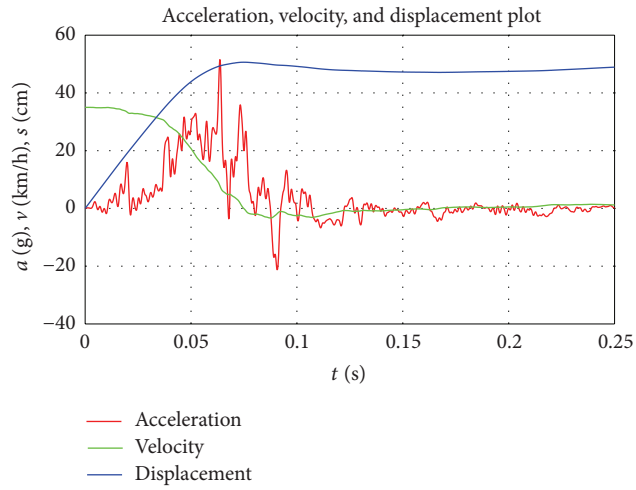


Figure 1: Crash test data.

(i) $m = 873$ kg;

(ii) $s(t = 0) = 0$ m;

(iii) $v(t = 0) = v_0 = 9.7222$ m/s,

where the time $t = 0$ shows the time of impact, s is the displacement of the vehicle, and v is the velocity of the vehicle. The results can be seen in Figure 1, where the red line shows the acceleration in x direction in g (-9.81 m/s^2), the blue line shows displacement in cm, and the green line shows velocity in km/h. These non-SI data types are used to scale the figure to fit all three graphs on one figure with a single y-axis.

3. Mathematical Models

3.1. Spring-Mass Model. The model shown in Figure 2 is based on the Kelvin model in [7, 19]. In this model, the vehicle is modeled as a mass and the crashing effect is modeled as a linear spring. The usage of a linear spring could be justified if the metal in the car body was to follow Hooke's Law, but since energy is absorbed in the crash, that is not the case. Therefore the results may not be satisfactory. In order to identify the constant spring coefficient, k, some terms need to be defined and found in the experimental crash data:

t_m: time of maximum dynamic crush, $v(t_m) = 0$;

C: vehicle dynamic crush at time t_m, $C = s(t_m)$;

t_c: the time at the geometric center of the crash pulse from time zero to t_m.

The centroid time t_c can be obtained by either integrating the acceleration to find the geometric center or using the following formula:

$$t_c = \frac{s(t_m)}{v(0)} = \frac{C}{v_0}. \quad (1)$$

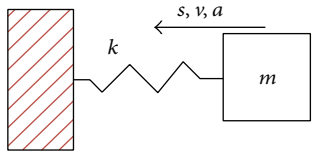


Figure 2: A spring-mass model.

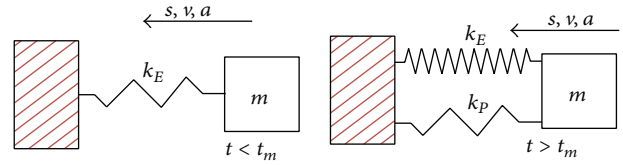


Figure 3: Elastoplastic model.

The normalized centroid time and angular position at dynamic crush is defined as [19]

$$\tau_c = t_c\omega_e = \left(\frac{C}{v_0}\right)\omega_e = e^{-\zeta\tau_m}, \tag{2}$$

$$\tau_m = t_m\omega_e = \frac{1}{\sqrt{1-\zeta^2}}\arctan\frac{\sqrt{1-\zeta^2}}{\zeta}, \tag{3}$$

where ω_e is the natural frequency of the system and ζ is the damping ratio. By combining (2) and (3), the following relation between the centroid time and time of maximum dynamic crush is found:

$$\frac{\tau_c}{\tau_m} = \frac{t_c}{t_m} = \frac{\sqrt{1-\zeta^2}}{\arctan\left(\sqrt{1-\zeta^2}/\zeta\right)}e^{\left[\left(-\zeta/\sqrt{1-\zeta^2}\right)\arctan\left(\sqrt{1-\zeta^2}/\zeta\right)\right]}. \tag{4}$$

By solving (4) with respect to ζ, the damping of the system is determined. The natural frequency of the system (ω_e) is determined by solving (3) with respect to the natural frequency as follows:

$$\omega_e = \frac{1}{t_m\sqrt{1-\zeta^2}}\arctan\frac{\sqrt{1-\zeta^2}}{\zeta}. \tag{5}$$

If ζ is 0 or $t_c/t_m > 2/\pi$, Huang [7] proposed a method for undamped systems. The system is reduced to a mass-spring model, and a new time of maximum crush is calculated as follows:

$$t_m = \frac{\pi}{2}t_c. \tag{6}$$

The spring and damper constants (k and c, resp.) can be found by using ζ and ω_e in two second-order system equations, respectively, as follows:

$$k = \omega_e^2 m, \tag{7}$$

$$c = 2\omega_e\zeta m. \tag{8}$$

3.2. Elastoplastic Spring. The model shown in Figure 3 is based on an elastoplastic model in [8]. k_E stands for elastic spring constant and k_P is used for the plastic deformation in the spring. Since the car body does not follow Hooke's Law through the entire crash, some of the energy in the mass needs to be absorbed in plastic deformations. This is modeled by adding a much stiffer spring after the car has crashed its full length into the pole. However since springs do not remove any energy in the system, the results may not be satisfactory. In order to identify the two spring constants that fit the crash data, some terms need to be defined and located in the crash data:

d_e: maximum rebound deflection after crash;

v': maximum rebound velocity.

Based on energy considerations, the following equations are set up:

$$\Delta E = \frac{1}{2}k_E C^2 = \frac{1}{2}mv_o^2, \tag{9}$$

$$\Delta E' = \frac{1}{2}k_P d_e^2 = \frac{1}{2}mv'^2, \tag{10}$$

where ΔE is the energy before the crash. $\Delta E'$ is the energy after the crash. Although the deflection energy should include the elastic spring coefficient k_E, it has been neglected because $k_P \gg k_E$. The maximum force on the vehicle makes a relationship between the elastic and plastic spring coefficients as follows:

$$F = k_E C = k_P d_e, \tag{11}$$

$$\text{or } \frac{C}{d_e} = \frac{k_P}{k_E}. \tag{12}$$

COR (coefficient of restitution) is defined as the percentage of the initial energy that is restored after the crash. This ratio can be found by

$$\text{COR}^2 = \frac{\Delta E'}{\Delta E} = \left(\frac{v'}{v_0}\right)^2 = \frac{k_P d_e^2}{k_E C^2}. \tag{13}$$

The elastic spring coefficient can be found by using (9) and the plastic spring coefficient is determined by substitution (12) into (13):

$$k_E = m\frac{v_0^2}{C^2}, \tag{14}$$

$$k_P = \frac{k_E}{\text{COR}^2}. \tag{15}$$

3.3. Maxwell Spring-Damper Model. This model is based on the Maxwell Spring-Damper model shown by Huang in [7]. The mass is connected to the wall with a spring and a damper in series. This model is suitable for material responses that exhibit relaxation and creep, a time dependent phenomena.

In vehicle impact modeling, it is suited for the localized impact, where the vehicle effective stiffness is low [7]. The system is described with the following differential equations:

$$m'\ddot{x}' = -x'k - \left(\dot{x}' - \dot{x}\right) \cdot c \tag{16}$$

$$m\ddot{x} = \left(\dot{x}' - \dot{x}\right) \cdot c, \tag{17}$$

where x and x' are the displacement by the masses m and m', respectively; see Figure 4. By differentiating (16) and (17) with respect to time and setting $m' = 0$, one has

$$\begin{aligned} 0 &= -\dot{x}'k - \left(\ddot{x}' - \ddot{x}\right) \cdot c, \\ m\frac{d\ddot{x}}{dt} &= \left(\ddot{x}' - \ddot{x}\right) \cdot c. \end{aligned} \tag{18}$$

From (18), we obtain

$$\dot{x}' = -\frac{m}{k} \cdot \frac{d\ddot{x}}{dt}. \tag{19}$$

By inserting (19) into (17), a differential equation without the massless displacement x' is formed; see (20), and its characteristic equation is given by (21).

Consider

$$m\ddot{x} = \left(-\frac{m}{k} \cdot \frac{d\ddot{x}}{dt} - \dot{x}\right) \cdot c, \tag{20}$$

$$r^3 + \frac{k}{c}r^2 + \frac{k}{m}r = 0. \tag{21}$$

Equation (21) can be solved in order to find the roots of the equation. However, the coefficients k and c are unknown, and therefore one cannot tell whether the roots contain imaginary numbers. Two solutions are therefore proposed: one for real roots and one for imaginary roots. For the real roots, the roots are defined as

$$\begin{aligned} r_1 &= 0, \\ r_2 &= a + b, \\ r_3 &= a - b, \end{aligned} \tag{22}$$

where

$$\begin{aligned} a &= -\frac{k}{2c}, \\ b &= \sqrt{\left(\frac{k}{2c}\right)^2 - \frac{k}{m}}. \end{aligned} \tag{23}$$

If the roots are imaginary, the following roots are defined:

$$\begin{aligned} r_1 &= 0, \\ r_2 &= a + ib, \\ r_3 &= a - ib, \end{aligned} \tag{24}$$

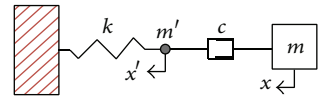


Figure 4: Maxwell spring-damper model.

where

$$\begin{aligned} a &= -\frac{k}{2c}, \\ b &= \sqrt{\frac{k}{m} - \left(\frac{k}{2c}\right)^2}. \end{aligned} \tag{25}$$

The solution for the displacement of the mass can be written as two equations, each depends on whether the roots are real or imaginary.

If real roots, then

$$x(t) = C_1 + C_2 e^{(a+b)t} + C_3 e^{(a-b)t}; \tag{26}$$

if imaginary roots, then

$$x(t) = C_1 + e^{at}\left[C_2 \cos(bt) + C_3 \sin(bt)\right], \tag{27}$$

where C_1, C_2, and C_3 are arbitrary constants. Now that the shapes of the different solutions are known, one can fit the equations to the displacement of the experimental data to determine the unknown variables C_1, C_2, C_3, a, and b. Once these variables are found, the spring and damper coefficients can be found by using the equations for a and b.

3.4. Nonlinear Spring and Damper. The model shown in Figure 5 represents the vehicle with a nonlinear spring and damper that crashes into an obstacle. The purpose of using a nonlinear spring and damper is to make the simulated crash act nonlinear based on the speed and position of the car. To justify the use of nonlinear coefficients in the crash, the crash data shown in Figure 1 shows a velocity curve that matches a mass-spring system before the time of maximum crush. However, right after the crash, the system is damped critically to almost a tenth of the initial velocity. With a linear damper, the velocity of the simulated system would decrease rapidly because of the high initial velocity. But with a nonlinear damper, the shape of the damping coefficient can be controlled in order to dampen the car less during high velocities and critically dampen it when the velocity is small. Using a nonlinear spring and damper is justified from a material science perspective, since most true-stress-true-strain curves show high nonlinearities from initial stretch to fracture.

The nonlinear spring and damper parameters are identified with the help of computational power in an optimization algorithm. MATLAB is used to create an optimization routine based on a predefined shape of the nonlinear spring and damper coefficients, an objective, and side-constraints. Figure 6 shows the predefined shape of the nonlinear spring and damper.

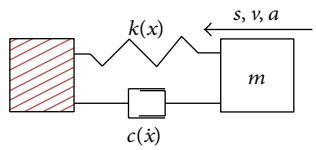


Figure 5: Nonlinear spring and damper model.

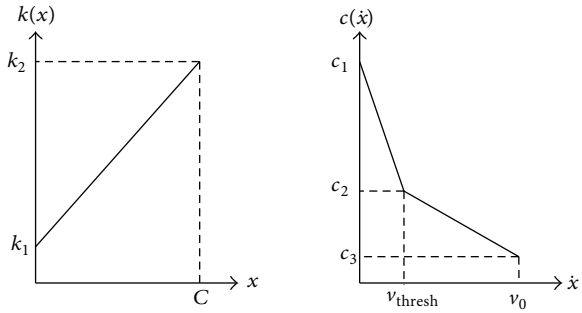


Figure 6: The predefined shape of the nonlinear spring and damper.

The unknown variables k_1, k_2, c_1, c_2, c_3, and v_{thresh} are determined by an interior-point optimization algorithm explained in Section 4.1 to fit the crash data best. The function that is chosen to be minimized is the norm of the absolute error between the displacement of the simulated crash and the experimental crash data:

$$[\text{Error}] = |\vec{x} - \vec{s}|^T |\vec{x} - \vec{s}|, \tag{28}$$

where $\vec{x}$ is a vector containing displacement of the simulated data and $\vec{s}$ is a vector containing displacement of the experimental crash data. The last part of an optimization routine is to set side-constraints for the program to follow. Side-constraints are split into two functions: functions to keep smaller or equal to zero and functions to keep equal to zero. These side-constraints will help the program to keep the unknown variables within a threshold and to make the end result match satisfactory. The side-constraints that are used are shown in Table 1.

Many of the side-constraints are used to keep the predefined shape of the nonlinear spring and damper and to keep the damper and spring coefficients as positive values throughout the simulation. The side-constraints for x and $\dot{x}$ are used to match the time and displacement at maximum dynamic crush in the crash data. When the objective function and the side-constraints are determined, the program can start and the optimization routine will find the best value for the unknown variables that fits the crash test data and stays within the constraints.

3.5. Nonlinear Spring and Damper Model II. The first nonlinear spring and damper model shown in Section 3.4 can be improved upon to reach a better result by adding more breakpoints into the spring and damper functions. However, more parameters give any optimization routine a larger amount of local minima, and a global minimum is harder to find. An algorithm with a wide search area is used to try and find local minima, specifically the firefly algorithm described in Section 4.2.

Table 1: Side-constraints.

$k_2 \geq k_1$	$c_1 \geq c_2$	$c_1 \geq c_3$
$c_2 \geq c_3$	$v_{\text{thresh}} \leq v_0$	$v_{\text{thresh}} \geq 0$
$x(t = t_m) = C$	$\dot{x}(t = t_m) = 0$	$k_1, k_2, c_1, c_2, c_3 \geq 0$

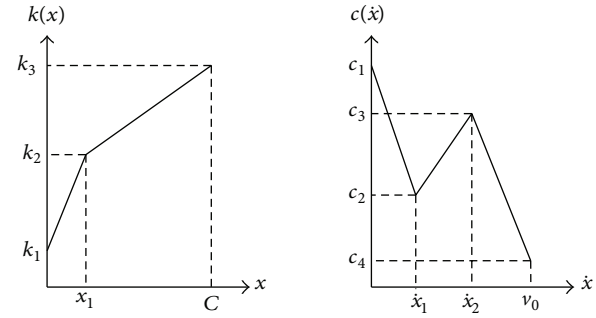


Figure 7: Spring and damper coefficient visualization.

The model that is proposed to be optimized against the experimental data has more breakpoints than the last model, and it is shown in Figure 7. Compared to the model shown in Figure 6, this model has two extra breakpoints: one in the damper curve and one in the spring curve. This adds complexity to the model, but also the accuracy of the results. In addition to having more breakpoints than the previous model, the optimization was done using fewer constraints. The constraints that were forcing the shape of the curve were removed in order to create a wider area of parameters to optimize in. The only constraints set for this model is that all of the parameters needs to be larger than *zero*, and the breakpoints are within the given largest displacement or initial velocity.

The function that is to be minimized is the norm of the error between the experimental crash data and simulated crash data:

$$[\text{Error}] = |\vec{x} - \vec{s}|^T |\vec{x} - \vec{s}|, \tag{29}$$

where $\vec{x}$ is a vector containing displacement of the simulated data and $\vec{s}$ is a vector containing displacement of the experimental crash data.

4. Optimization Algorithms

4.1. Interior-Point Method. An interior-point optimization algorithm is a deterministic algorithm that finds a vector that minimizes a given objective function with respect to linear and nonlinear constraints. This algorithm was invented by John von Neuman in 1873. The algorithm can be used to solve linear and nonlinear convex optimization problems with equality and inequality constraints. The algorithm is deterministic because it follows a set of rules to find local/global minima. This means that every time this algorithm is used with the same initial values, the algorithm will find the same

results. This algorithm will be used to find parameters for a nonlinear mass-spring model in Section 3.4.

4.2. Firefly Algorithm. The firefly optimization algorithm is a nature inspired metaheuristic optimization algorithm that was invented in 2008 by Yang and He [10]. This algorithm can be used to find good solutions for linear and nonlinear optimization problems with equality and inequality constraints. Metaheuristic algorithms can be used to find optimal solutions to a variety of problems, but compared to a deterministic algorithm, a metaheuristic algorithm cannot guarantee to find global or a local minima. This particular algorithm implements some form of stochastic variables by means of random initial parameter values and random walks. The consequence is that finding a local or a global minima could be time consuming or even impossible for a given problem. This algorithm is based on three principles for real life fireflies [10].

(i) Fireflies are unisex so that one firefly will be attracted to all other fireflies.

(ii) The attractiveness is proportional to the brightness, and they both decrease as their distance increases. Thus for any two flashing fireflies, the less bright one will move towards the brighter one. If there is no brighter one than a particular firefly, it will move randomly.

(iii) The brightness of a firefly is determined by the landscape of the objective function.

In this algorithm, each firefly represents a model with a set of parameters for the objective function. As these fireflies move around in the "parameter field," the objective function changes and their attractiveness changes with it. The equation for the movement of the fireflies and a further explanation can be found in [10].

5. Simulation Results

5.1. Spring-Mass Model. The experimental data shown in Figure 1 is used in order to produce the results. The data extracted from the plot is

$$\begin{aligned} t_m &= 0.0749\,\text{s} \\ t_c &= 0.0520\,\text{s} \\ C &= 0.5063\,\text{m} \\ \frac{t_c}{t_m} &= 0.6951. \end{aligned} \tag{30}$$

The relation between t_c and t_m gives a number larger than $2/\pi$ and therefore the system is reduced to a mass-spring system as shown in Figure 2. The new time of dynamic crush is calculated by using (6) as $t_m = 0.0818$ s. Since the damping ζ is zero, the natural frequency is calculated by using (5) as $\omega_e = 19.2028$ rad/s. And the spring coefficient can finally be determined by using (7):

$$k = \omega_e^2 m = 321,535\,\text{N/m}. \tag{31}$$

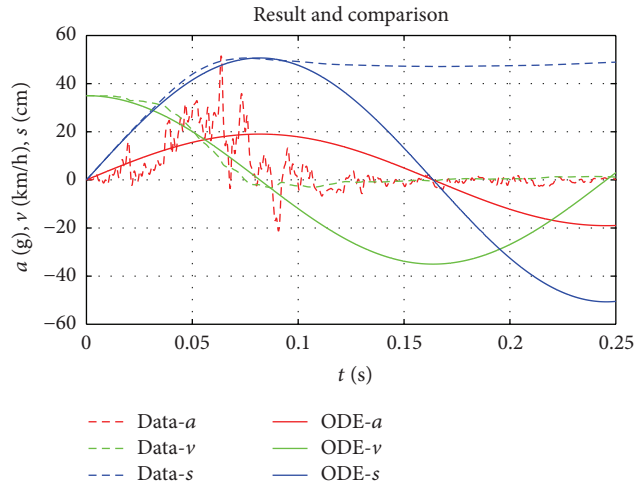


Figure 8: Results and comparison between Kelvin model and crash data.

A comparison between the Kelvin model and the experimental data is shown in Figure 8, where the stapled lines indicate the experimental data and the continuous lines indicate the simulated response.

5.2. Elastoplastic Spring. Using the crash data shown in Figure 1, these data were extracted as follows:

$$\begin{aligned} C &= 0.5063\,\text{m}, \\ v' &= 0.9183\,\text{m/s}. \end{aligned} \tag{32}$$

The COR is calculated from (13) to be COR = 0.0945 and the elastic and plastic spring coefficients are determined by using (14) and (15), respectively, as follows:

$$\begin{aligned} k_E &= 873 \cdot \frac{10^2}{0.5063^2} = 321,944\,\text{N/m}, \\ k_P &= \frac{321.9}{0.0945^2} = 36,084,417\,\text{N/m}. \end{aligned} \tag{33}$$

Results and comparison between the experimental crash data and the simulated response are shown in Figure 9, where the stapled lines indicate the experimental data and the continuous lines indicate the simulated response.

5.3. Maxwell Spring-Damper Model. The displacement of the experimental crash data was used in a Curve Fitting Toolbox in MATLAB in order to find the unknown coefficients C_1, C_2, C_3, a, and b for (26) and (27). The curve fitting for the real roots can be seen in Figure 10 and the curve fit for the imaginary roots in Figure 11 and Table 2 shows the results from the curve fitting and their corresponding damper and spring coefficients.

Obviously, the values for k and c extracted from the real roots curve fit are wrong because they have negative values. On the other hand, the damper and spring coefficients found using the curve for imaginary roots are valid. The result

Table 2: Damper and spring constants for Maxwell model.

Parameter	Real roots	Imaginary roots
a	−14.5	−33.42
b	−14.6	39.02
k	−2,540 N/m	2,304,245 N/m
c	−87.6 Ns/m	34,474 Ns/m

Table 3: Parameter values.

Parameter	Value	Unit
c_1	568,160	Ns/m
c_2	39,956	Ns/m
c_3	179.50	Ns/m
v_{thresh}	0.7948	m/s
k_1	237	N/m
k_2	253,440	N/m

Table 4: Linear bounds for the model.

Parameter	Lower bound	Upper bound
k_1	0	1,000,000
k_2	0	1,000,000
k_3	0	1,000,000
x_1	0	C
c_1	0	1,000,000
c_2	0	1,000,000
c_3	0	1,000,000
c_4	0	1,000,000
$\dot{x}_1$	0	v_0
$\dot{x}_2$	0	v_0

of the time integration using the spring and damper and a comparison to the crash data can be seen in Figure 12.

5.4. Nonlinear Spring and Damper. By using an *interior-point* solver in MATLAB with the objective and constraint functions given in Section 3.4, the algorithm returned the values shown in Table 3 for the unknown parameters. A comparison of the optimized system versus the experimental data is shown in Figure 13, where the stapled lines are given by the experimental data and the continuous lines are the results from the time simulation.

5.5. Nonlinear Spring and Damper Model II. The nonlinear model shown in Section 3.5 was optimized by using the firefly algorithm explained in Section 4.2. The initial parameter values for the fireflies were found by choosing random values between the upper and lower bounds shown in Table 4. For this optimization, 40 fireflies were used with a randomness of $\alpha_0 = 0.005$, a cooling factor of $\delta = 0.5^{(1/500)}$, attractiveness at zero distance $\beta_0 = 1$, and a scaling factor $\gamma = 1/\sqrt{L}$, where L is the norm of the difference between the upper bounds and the lower bounds. Since the scale of the problem is quite large, this metaheuristic algorithm struggled

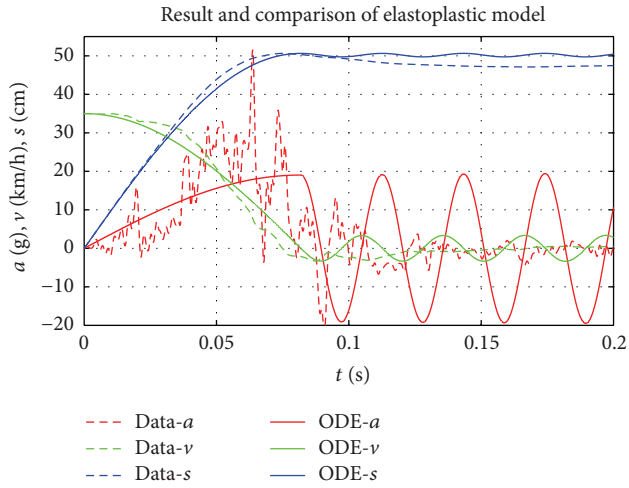


Figure 9: Results and comparison elastoplastic model.

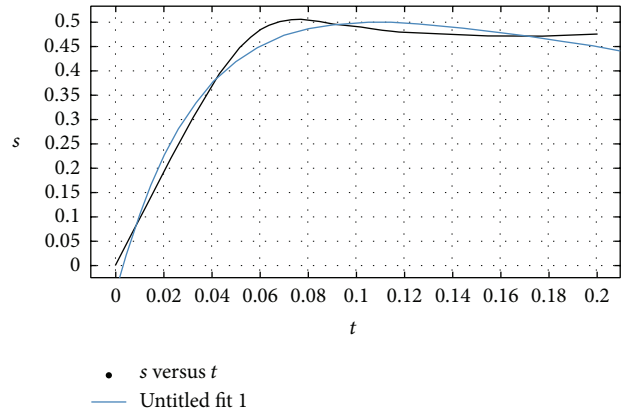


Figure 10: Curve fit with real roots.

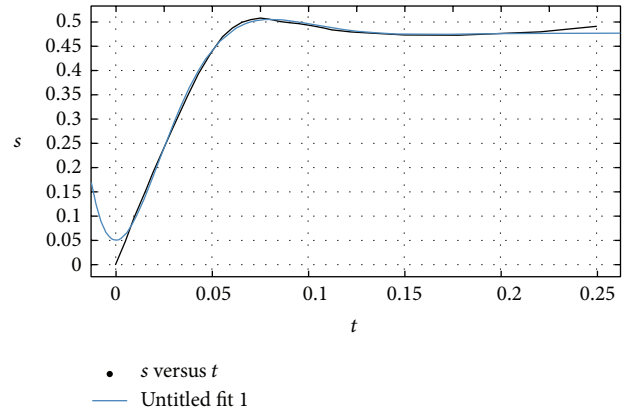


Figure 11: Curve fit with imaginary roots.

to find a good solution. It struggled because the fireflies may easily clump together in nonoptimal minima, or even worse, not find any local minima at all. Because of this, the algorithm had to be run multiple times to find a good solution to the problem. This is simply the weakness of metaheuristic algorithms with stochastic terms. A reasonably good solution

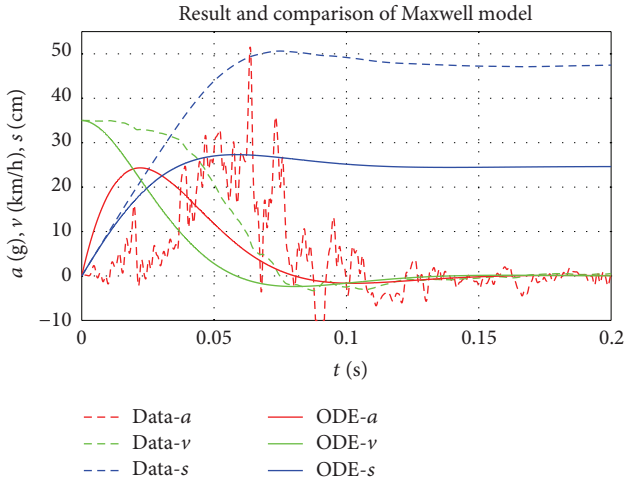


Figure 12: Results and comparison Maxwell model.

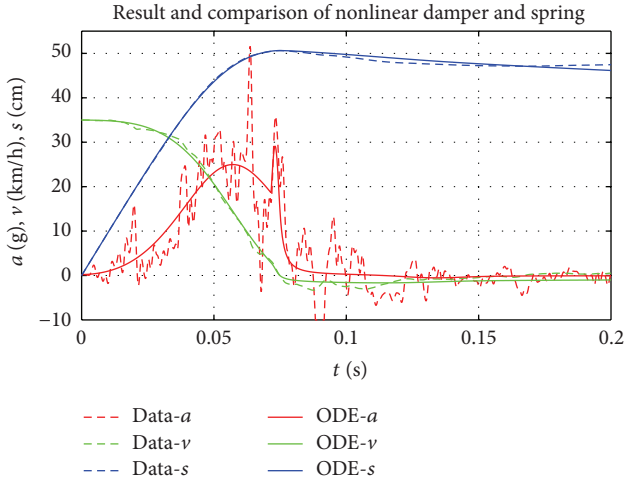


Figure 13: Results and comparison between crash data and simulated data.

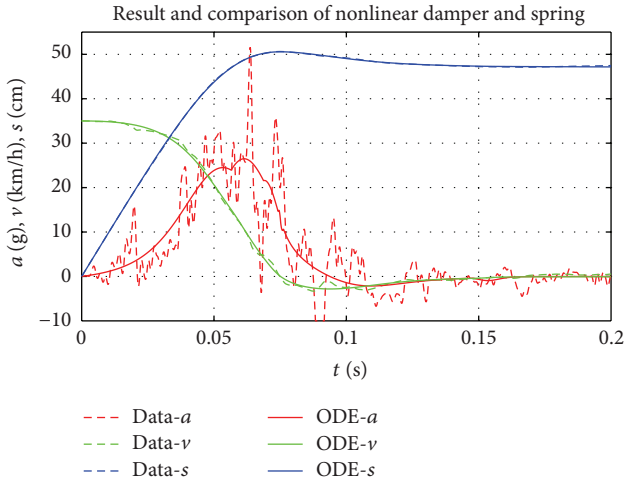


Figure 14: Results and comparison between optimized model and experimental crash data.

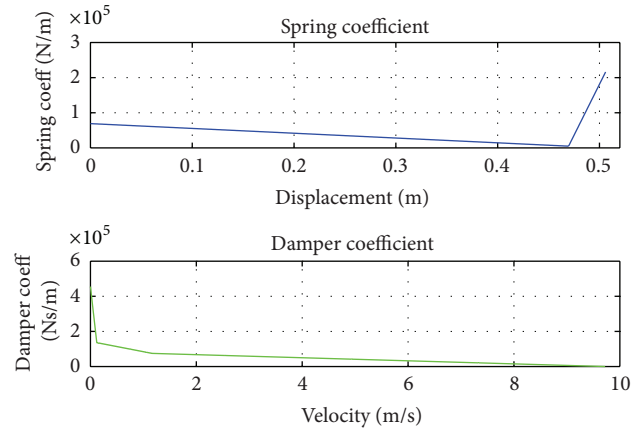


Figure 15: The nonlinear damper and spring characteristics.

Table 5: Parameters found through the Firefly optimization.

Parameter	Value	Unit
k_1	82,357	N/m
k_2	4	N/m
k_3	197,519	N/m
x_1	0.472	m
c_1	86,585	Ns/m
c_2	80,056	Ns/m
c_3	18,721	Ns/m
c_4	81	Ns/m
$\dot{x}_1$	2.55	m/s
$\dot{x}_2$	7.23	m/s

was found in the end, and the parameters for this model are shown in Table 5. A comparison of the optimized system versus the experimental data is shown in Figure 14, where the stapled lines are given by the experimental data and the continuous lines are the results from the time simulation. Also, in Figure 15 a representation of the non-linear spring and damper coefficients are shown versus the absolute value of the displacement and velocity of the vehicle, respectively.

6. Conclusion

The Kelvin model in Section 3.1 that consisted of a mass, a spring, and a damper showed good results prior to the time of maximum crush. The reason for this is due to the removal of the damping forces, and the model was therefore reduced to a spring-mass model. As there is no damping, the system will oscillate for infinite time with the maximum crush displacement as the amplitude. This infinite oscillation makes the time of interest in this model be within $t \in [0, t_m]$.

The elastoplastic model in Section 3.2 shows mixed but good results. The first part, the time before maximum crush, shows similarities to the Kelvin model. Both models are undamped and therefore the curves are similar. However, right after the time of maximum crush, the theory of a plastic damaged spring kicks in and makes the system oscillate

around the maximum crush displacement. This gives the system a relatively stable displacement after maximum crush, but like the Kelvin model, the oscillation will never stop and therefore the time of interest should be within $t \in [0, 0.1\ \text{s}]$.

The Maxwell model in Section 3.3 showed great potential in its description that it would fit a physical car crash. The results however lacked good results. The damper kicked in too early and damped the response. This makes the time integrated displacement curve go halfway to the maximum crush. However when looking at the shape of the simulated displacement curve, one can see that it resembles the shape of the crash data quite well. The possible reasons for the bad results may come from inaccurate curve fitting or other faults. Either way, the Maxwell model shows potential for a crash test in its shape and should therefore be further investigated.

The nonlinear spring and damper model shown in Section 3.4 showed good results with small amounts of error when comparing the simulated data and the crash data. One part that could be improved is the displacement of the model when time reaches infinite. Because of the spring forces that are active near the end of the simulation, the displacement of the mass will converge to zero.

The second nonlinear spring and damper model shown in Section 3.5 gave even better results than the first model, mostly because of the extra mobility given by the extra breakpoints. One thing to note is that the spring forces are almost zero at the end of the simulation, and this makes the displacement stay at the end position when time reaches infinite.

It is noted that the extensions of the proposed method to the vehicle crash modeling with vehicle dynamics including active suspension control system [20, 21] deserve further investigation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the grant of the collaborative research project between the University of Agder and Telemark University College.

References

[1] G. Sun, F. Xu, G. Li, and Q. Li, "Crashing analysis and multiobjective optimization for thin-walled structures with functionally graded thickness," *International Journal of Impact Engineering*, vol. 64, pp. 62–74, 2014.

[2] Y. Peng, J. Yang, C. Deck, and R. Willinger, "Finite element modeling of crash test behavior for windshield laminated glass," *International Journal of Impact Engineering*, vol. 57, pp. 27–35, 2013.

[3] H. Al-Thairy and Y. C. Wang, "A simplified analytical method for predicting the critical velocity of transverse rigid body impact on steel columns," *International Journal of Impact Engineering*, vol. 58, pp. 39–54, 2013.

[4] B.-Y. Kim, C.-M. Jeong, S.-W. Kim, and M.-W. Suh, "A study to maximize the crash energy absorption efficiency within the limits of crash space," *Journal of Mechanical Science and Technology*, vol. 26, no. 4, pp. 1073–1078, 2012.

[5] X. Liao, Q. Li, X. Yang, W. Zhang, and W. Li, "Multiobjective optimization for crash safety design of vehicles using stepwise regression model," *Structural and Multidisciplinary Optimization*, vol. 35, no. 6, pp. 561–569, 2008.

[6] W. Pawlus, H. R. Karimi, and K. G. Robbersmyr, "Investigation of vehicle crash modeling techniques: theory and application," *The International Journal of Advanced Manufacturing Technology*, vol. 70, no. 5–8, pp. 965–993, 2014.

[7] M. Huang, *Vehicle Crash Mechanics*, CRC Press, Boca Raton, Fla, USA, 2002.

[8] W. Pawlus, H. R. Karimi, and K. G. Robbersmyr, "Mathematical modeling of a vehicle crash test based on elasto-plastic unloading scenarios of spring-mass models," *International Journal of Advanced Manufacturing Technology*, vol. 55, no. 1–4, pp. 369–378, 2011.

[9] A. Elmarakbi, M. Elkady, and H. El-Hage, "Development of a new crash/dynamics control integrated mathematical model for crashworthiness enhancement of vehicle structures," *International Journal of Crashworthiness*, vol. 18, no. 5, pp. 444–458, 2013.

[10] X. Yang and X. He, "Firefly algorithm: recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, pp. 36–50, 2013.

[11] L. Guo and G. Wang, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 696491, 21 pages, 2013.

[12] L. Tang, H. Wang, G. Li, and F. Xu, "Adaptive heuristic search algorithm for discrete variables based multi-objective optimization," *Structural and Multidisciplinary Optimization*, vol. 48, no. 4, pp. 821–836, 2013.

[13] L. Zhao, W. Pawlus, H. R. Karimi, and K. G. Robbersmyr, "Data-based modeling of vehicle crash using adaptive neural-fuzzy inference system," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 684–696, 2014.

[14] W. Pawlus, H. R. Karimi, and K. G. Robbersmyr, "Data-based modeling of vehicle collisions by nonlinear autoregressive model and feedforward neural network," *Information Sciences*, vol. 235, pp. 65–79, 2013.

[15] H. R. Karimi, W. Pawlus, and K. G. Robbersmyr, "Signal reconstruction, modeling and simulation of a vehicle full-scale crash test based on Morlet wavelets," *Neurocomputing*, vol. 93, pp. 88–99, 2012.

[16] W. Pawlus, H. R. Karimi, and K. G. Robbersmyr, "Reconstruction and simulation of the vehicle to road safety barrier oblique collision based on the Levenberg-Marquardt algorithm," *International Journal of Crashworthiness*, vol. 17, no. 6, pp. 676–692, 2012.

[17] W. Pawlus, K. G. Robbersmyr, and H. R. Karimi, "Mathematical modeling and parameters estimation of a car crash using data-based regressive model approach," *Applied Mathematical Modelling*, vol. 35, no. 10, pp. 5091–5107, 2011.

[18] K. G. Robbersmyr, "Calibration test of a standard Ford Fiesta 1.1L, model year 1987, according to NS-EN 12767," Tech. Rep. 43/2004, Agder Research, Grimstad, Norway, 2004.

[19] W. Pawlus, J. E. Nielsen, H. R. Karimi, and K. G. Robbersmyr, "Mathematical modeling and analysis of a vehicle crash," in

Proceedings of the 4th European Computing Conference (ECC '10), pp. 194–199, Stevens Point, Wis, USA, April 2010.

[20] H. Li, X. Jing, and H. R. Karimi, "Output-feedback-based H_∞ control for vehicle suspension systems with control delay," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 436–446, 2014.

[21] W. Sun, Y. Zhao, J. Li, L. Zhang, and H. Gao, "Active suspension control with frequency band constraints and actuator input delay," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 530–537, 2012.