PREDICTING THE BEHAVIOR OF THE VEHICLE IN A FRONTAL COLLISION INTO RIGID BARRIER BY USING A SYSTEM IDENTIFICATION TOOLBOX

JOZEF KMEC[1], JOZEF PAVELKA[1], JAROSLAV SOLTES[1]

[1]Faculty of Humanities and Natural Sciences, Presov University, Prešov, Slovakia

DOI : 10.17973/MMSJ.2019_12_2019035

e-mail: jozef.kmec@unipo.sk

The main objective of this paper is to establish models of measured real-time data that are obtained in a frontal impact of a vehicle into the rigid barrier. When it tcomes to modeling the vehicle crash we can distinguish two approaches. The first one utilizes CAE (Computer Aided Engineering) software including FEA (Finite Element Analysis) while the second one is based on the System Identification Toolbox , which provides MATLAB® functions, Simulink® blocks, and an app for constructing the models of dynamic systems from measured input-output data. The toolbox also provides algorithms for embedded online parameter estimation.

KEYWORDS

Matlab, System identification toolbox, State-space model, Process model with transfer function, Polynomial model, Transfer function model.

1 INTRODUCTION

The vehicle crash test is usually done in order to ensure safe design standards . Due to advanced research in simulation software, simulated crash tests can be performed and evaluated by the full-scale crash test. Therefore, cost associated with real crash test can be reduced. System identification concerns construction and validation of mathematical models from dynamic input / output data. In experiments the system reveals information about itself in terms of input and output measurements. System identification and tools for modeling are routinely used in industry [Munyazikwiye 2014]. There are several solutions available for the identification of mathematical models based on experimental test procedures. One of the most convenient and accessible solution is to use the system identification toolbox [Mathworks R2013b]. The system identification toolbox is largely based on the work of [Ljung 1999] and implements common techniques used in system identification. There is a substantial literature on the system identification [Ljung 1994]. The toolbox aids the user to fit both linear and nonlinear models to measured data sets known as black box modeling [Marzbanrad 2011b].

2 GETTING DATA FROM FRONTAL IMPACT TEST

Data for the System identification Toolbox were obtained from a frontal impact test into the rigid barrier with full coverage at a speed of 56.17 kmh^{-1} (15.88 ms^{-1}) according to NCAP (New Car Assessment Program) [NCAP 2017]. The tested vehicle was Honda Civic XL 2 door Coupe. The record was processed from an accelerometer that was firmly connected to the vehicle floor at the rear of the bodywork [Vlk 2003]. Accelerometer is a sensor of non-electric quantities, which converts the detected quantity into an electrical signal and is subsequently processed and evaluated. This electrical signal is characterized as a continuous analogue signal and then the A/D converter is sampled to discrete values. Recording from accelerometer at impact tests due to significant signal oscillation must be filtered by the CFC 60 filter (Channel Frequency Class) [Cichos 2006]. Tab. 1 records the most common type of filter used in this article.

Filter type	Filter parameters		Use the filter
CFC 60	3 dB limit frequency	100 Hz	Structure acceleration
	Stop damping	–30 dB	
	Sampling frequency	At least 600 Hz	

Table 1. Type of filter

In order to be able to measure impact tests, the signal processing must be performed under predetermined conditions. These regulations are laid down in SAE J211-1: Instrumentation for Impact Test, Part 1, Electronic Instrumentation. All quantities occurring in the impact test are specified in this standard.

3 SYSTEM IDENTIFICATION TOOLBOX

It lets you create and use models of dynamic systems not easily modeled from first principles or specifications. You can use time-domain and frequency-domain input-output data to identify continuous-time and discrete-time transfer functions, process models, and state-space models. In a dynamic system, the values of the output signals depend on both the instantaneous values of its input signals and also on the past behavior of the system. A model is a mathematical relationship between the system's input and output variables. Models of dynamic systems are typically described by differential or difference equations, transfer functions, state-space equations, and pole-zero-gain models. System Identification requires a model structure. A model structure is a mathematical relationship between input and output variables that contain unknown parameters. Examples of model structures are transfer functions with adjustable poles and zeros, state space equations with unknown system matrices, and nonlinear parameterized functions. The system identification process requires that you choose a model structure and apply the estimation methods to determine the numerical values of the model parameters.

4 ACHIEVED RESULTS AND THEIR DISCUSSION

In the Simulink, which is a graphical programming environment for modeling, simulating and analyzing multidomain dynamical systems, our model was created Fig.1. The main objective of this model is to process the measured data (unfiltered deceleration signal) so that at the output of the Scope 5 block we observe the course of deformation for the vehicle (Honda civic XL). The model consists of blocks that are in the library and can be moved to the model window. In addition to moving from the library, blocks in the model window can be copied in a standard manner. The name of the new block is automatically set so that it is unambiguous within the model window. To create our own model in Fig. 1, the following blocks were used: Signal From Workspace, Lowpass Filter, Scope, Workspace, Gain, Integrator, Add, Constant, and some of their copies. The simulation result, which is the deformation course and the velocity course of the vehicle, can be seen in the Scope 5 block, which is essentially Fig. 2 and the simulation itself is performed by the model shown in Fig. 1.

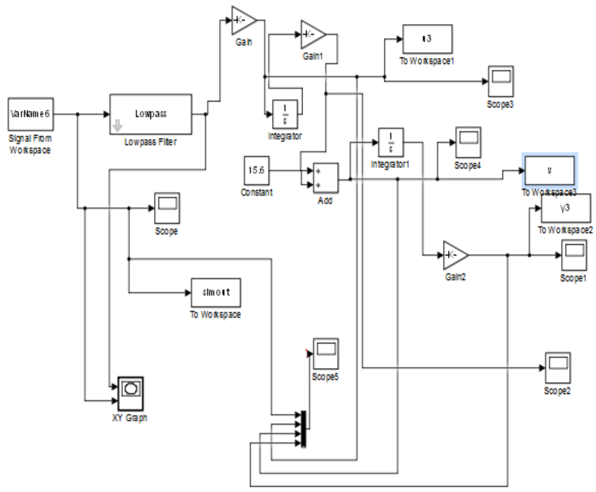

Figure 1. Own model in Simulink

In Fig. 2 (left down) is depicted the original data signal with noise (Deceleration in (g) vs. Time (s)) obtained from the accelerometer and the filtered data signal by the relevant filter CFC 60, see Fig. 2 (right down) (Deceleration (m/s2) vs. Time (s)). By integrating deceleration by time, it is possible to obtain a time-dependent velocity, see Fig. 2 (top right) (Velocity (m/s) vs. Time (s)). By double integration of deceleration by time, it is possible to calculate a time-dependent deformation , see Fig. 2 (top left) (deformation (m) vs Time (s)). From the measured data (record from accelerometer) and the processed data Fig. 1, it is possible to determine some parameters respecting the deceleration, velocity and deformation in a closed time interval (in our case it is 3 ms). For the vehicle Honda Civic XL, it is possible to determine maximum dynamic crush (max. deformation) C = 0.751 m at time t = 0.08s – from real dat, see Tab.2 and Fig. 2 [Vlk 2003, Evin 2016].

Real crash tests are difficult to realize – there is need for appropriate facilities, measuring devices, data acquisition process, qualified staff and of course – a car. Those factors make the test complicated, time – consuming and expensive enterprise. Therefore, instead of a real experiment it is justified to propose a mathematical model of a collision and analyze it to approximate its results. This allows us to predict the behavior of the real car without performing complicated crash tests.

Time of dynamic crush obtained from the models is exactly the same as in experiment: t = 0.08s. The maximum dynamic crush obtained from measured data (for Hondu Civic XL) is C = 0.751 m. Maximum dynamic crush C = 0.7646 m, which is obtained from the arx441 model is about 1.81% grather than that of the measured data. The greatest differences show arx441, arxqs models, which is in absolute numbers of 1.36 cm, see Tab.2.

	Time of dynamic crush t in (s)	Max. dynamic crush C (max. deformation) in (m)	Pole-Zero maps, distance from the zero (s^{-1})	Best fits
Measured data	0.08	0.751	-	-
tf1-model	0.08	0.751	|i0,009|, |-i0,009|	99.78%
ss1-model	0.08	0.7517	|0|	99.22%
P2U-model	0.08	0.7518	|-12|,|-95|	99.12%
n4s2-model	0.08	0.7503	|1|	98.97%
arx441-model	0.08	0.7645	|0.4+i0.4|,|0.4-i0.4|,|1|	94.71%
arxqs-model	0.08	0.7646	|0.4+i0.4|,|0.4-i0.4|,|1|	94.70%

Table 2. Gained data (both measured and calculated)

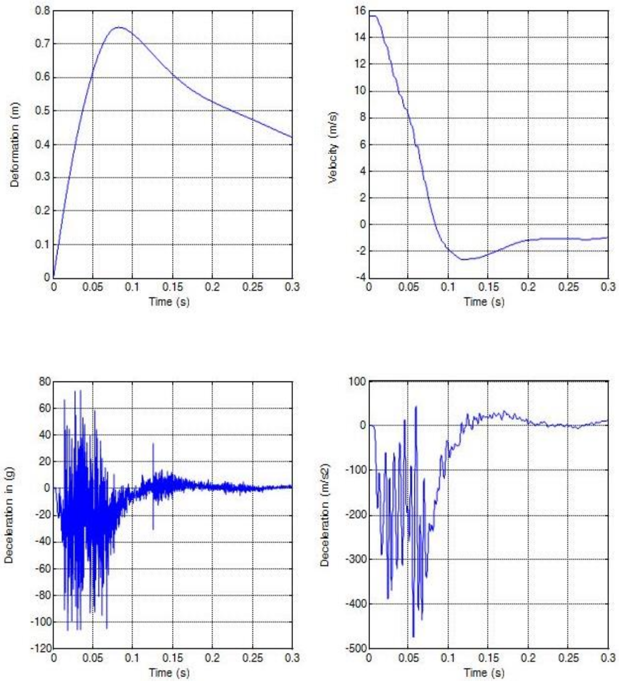

Figure 2. Oscilloscope outputs (Sccpe 5 block)

The processed data from Simulink´s own model , were further imported into the system identification toolbox as seen in Fig. 3 (input-deceleration output-deformation signals).

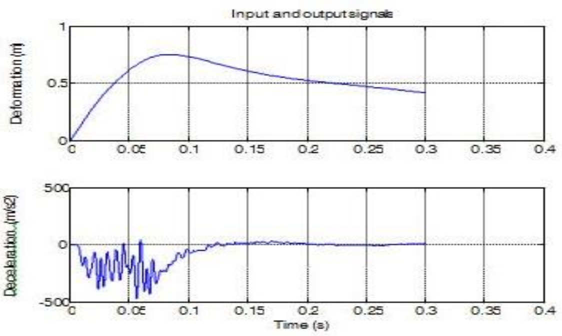

Figure 3. Input and output signals

4.1 Linear models using quick start

You can use the Quick Start feature in the System Identification app to estimate linear models. One of these is also the n4s2 - state-space model calculated using n4sid. The matlab algorithm automatically selects the model order (in this case, 2). n4sid - estimate state-space model using a subspace method and measured input-output data. This model is parametric and has the following structure:

$$dx/dt = Ax(t) + Bu(t) + Ke(t) \;, \quad y(t) = Cx(t) + Du(t) + e(t) \qquad (1)$$

where y*(t)* represents the output at time *t*, *u(t)* represents the input at time *t*, *x* is the state vector, and *e(t)* is the white-noise disturbance. The System Identification Toolbox product estimates the state-space matrices *A*, *B*, *C*, *D*, and *K*.

A =

	x1	x2
x1	0.9995	0.0004479
x2	-0.0004537	1

B =

	Deceleration
x1	4.975e-07
x2	2.179e-06

C =

	x1	x2
Deformation	16.96	-3.94

D =

	Deceleration
Deformation	0

K =

	Deformation
x1	0.05999
x2	0.1539

n4s2 - state-space model (pink line) is created from measured input-output data, see Fig. 4. Sample time: 0.0001 seconds. Deployment of poles and zeros, see Fig. 5.

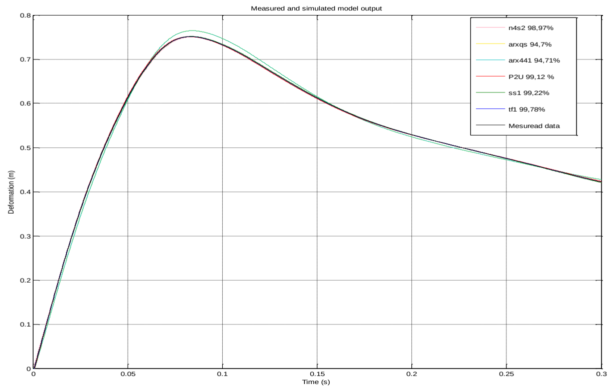

Figure 4. Measured and model outputs

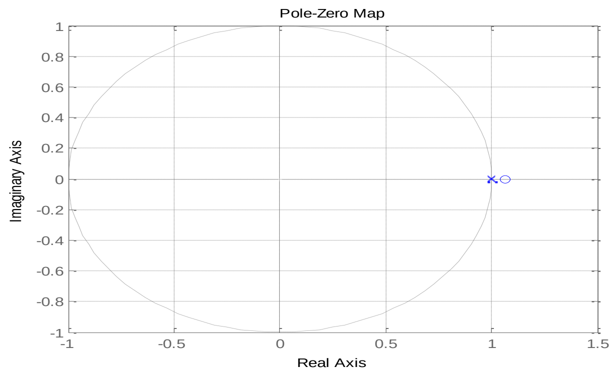

Figure 5. (Pole (cross)-Zero (circlet) map)

4.2 Transfer function model

The general transfer function model structure is:

$$Y(s) = (num(s)/den(s))U(s) + E(s) \;, \qquad (2)$$

where *Y*(*s*), *U*(*s*) and *E*(*s*) represent the Laplace transforms of the output, input and error, respectively. *num*(*s*) and *den*(*s*) represent the numerator and denominator polynomials that define the relationship between the input and the output. The roots of the denominator polynomial are referred to as the model *poles*. The roots of the numerator polynomial are referred to as the model zeros. You must specify the number of poles and zeros to estimate a transfer function model. The System Identification Toolbox product estimates the numerator and denominator polynomials, and input/output delays from the data. tf1 - model (blue line) is created from measured input-output data (From input "Deceleration" to output "Deformation"), see Fig. 4:

$$\frac{-0.0007561\,s + 0.9984}{s^2 + 0.01362\,s + 0.0001249} \qquad (3)$$

Parameterization: Number of poles: 2, Number of zeros: 1 , see Fig. 6

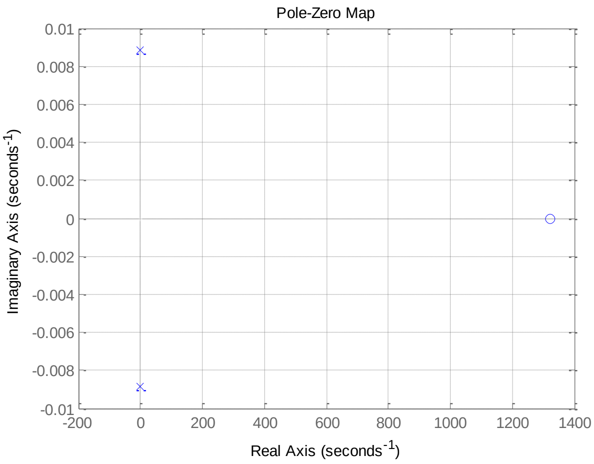

Figure 6. (Pole-Zero map)

4.3 State-space model

ss1- state-space model: (green line) created from measured input-output data, see Fig.4 [Munyazikwiye 2013]. Deployment of poles and zeros, see Fig. 7.

$$dx/dt = Ax(t) + Bu(t) + Ke(t)$$
$$y(t) = Cx(t) + Du(t) + e(t) \quad , \qquad (4)$$

where A =

	x1	x2	x3	x4
x1	1225	1624	-434	-1619
x2	-2.903e+w4	-1.063e+04	2444	6331
x3	-291.1	-60.77	15.06	12.81
x4	2.952e+04	1.084e+04	-2503	-6474

B =

	Deceleration
x1	-8.178e+06
x2	2.198e+07
x3	-5.649e+04
x4	-2.253e+07

C =

	x1	x2	x3	x4
Deformation	0.474	-18.81	-0.1272	-18.52

D =

	Deceleration
Deformation	0

K =

	Deformation
x1	689.2
x2	-1361
x3	1729
x4	921.8

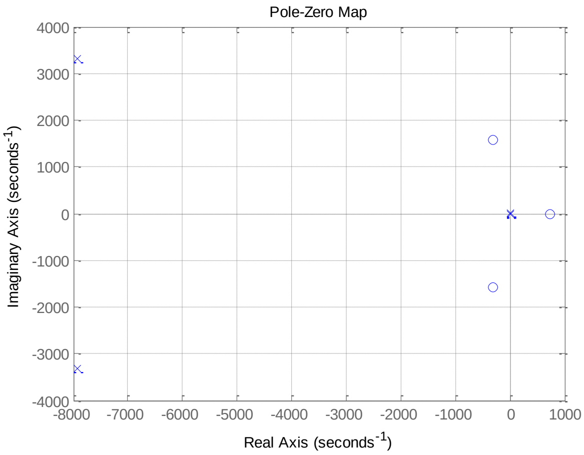

Figure 7. (Pole-Zero map)

4.4 About polynomial ARX and ARMAX models

For a single-input/single-output system (SISO), the arx model structure is:

$$y(t) + a_1 y(t-1) + \ldots + a_n y(t-n_a) =$$
$$b_1 u(t-n_k) + \ldots + b_{nb} u(t-n_k-n_b+1) + e(t)$$
(5)

y(t) represents the output at time *t*, *u(t)* represents the input at time *t*, n_a is the number of poles, n_b is the number of zeros plus 1, n_k is the input delay—the number of samples before the input affects the system output (called *delay* or *dead time* of the model), and *e(t)* is the white-noise disturbance. You specify the model orders n_a, n_b, and n_k to estimate arx models. The System Identification Toolbox product estimates the parameters $a_1 \ldots a_n$ ... and $b_1 \ldots b_n$ ... from the data.
Name: arx441 - model (turquoise line) is created from measured input-output data. Sample time: 0.0001 seconds , see Fig. 4. Deployment of poles and zeros, see Fig. 8 .

Discrete-time arx441 model:

$$A(z)y(t) = B(z)u(t) + e(t) \ , \qquad (6)$$

where A(z) = 1 - 2.77 z^-1 + 2.86 z^-2 - 1.39 z^-3 + 0.31 z^-4
B(z) = 3.864e-07 z^-1 - 1.006e-06 z^-2 + 8.404e-07 z^-3 - 2.14e-07 z^-4

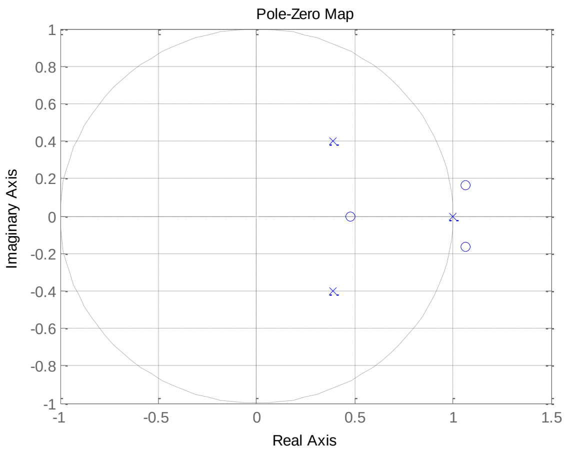

Figure 8. (Pole-Zero map)

arxqs - model (yelow line) is created from measured input-output data. Sample time: 0.0001 seconds. See Fig.4. Deployment of poles and zeros, see Fig. 9. Discrete-time arxqs model:

$$A(z)y(t) = B(z)u(t) + e(t) \quad , \qquad (7)$$

where A(z) = 1 - 2.776 z^-1 + 2.861 z^-2 - 1.394 z^-3 + 0.30 z^-4
B(z) = 3.1e-07 - 7.523e-07 z^-1 + 5.562e-07 z^-2 - 1.075e-07z^3

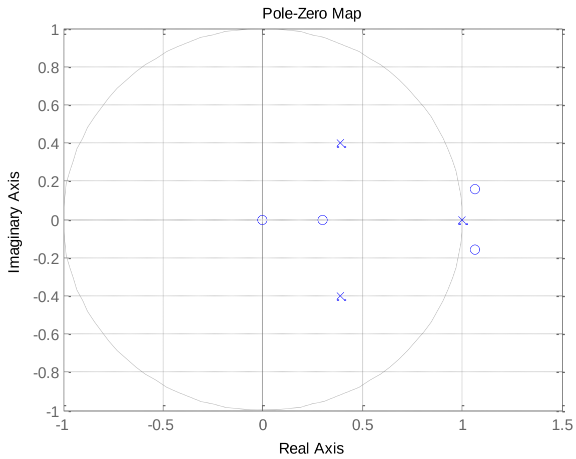

Figure 9. (Pole-Zero map)

4.5 Process model with transfer function

P2U - model (red line) is created from measured input-output data. Process model with transfer function is found in the formula:

$$G(s) = \frac{K_p}{1 + 2 * Zeta * Tw * s + (Tw * s)^2} \quad , \qquad (8)$$

where Kp = 8347.9
Tw = 90.677
Zeta = 0
Parameterization: 'P2U' , See Fig.4 . Deployment of poles and zeros, see Fig. 10 and Fig. 11

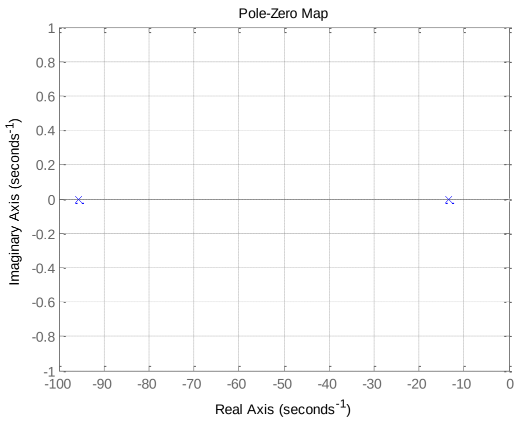

Figure 10. (Pole-Zero map)

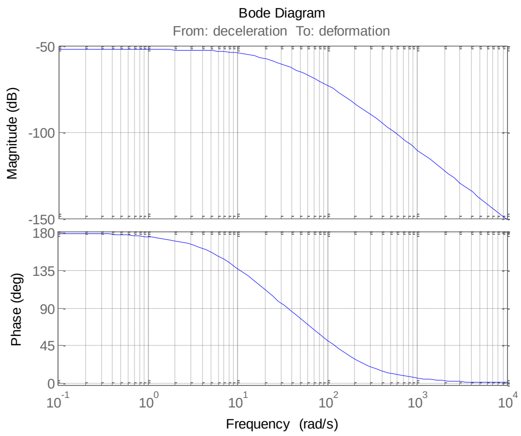

Figure 11. (Bode diagram)

5 CONCLUSION

Our own model Fig. 1 was created in Matlab-Simulink, which is capable of filtering off (Lowpass CFC 60 filter) the signal obtained from the accelerometer to improve modeling results.

Simulink's own model Fig. 1 allows to track the speed and deformation path over time (within 0.3 s), thus allowing to determine max. deformation from experimental data, max. deceleration of vehicle, max. deceleration at the head of the manikin, when the car's speed gets zero, at which time point the deceleration passes through a zero value and begins to acquire the positive values (separation of the vehicle from the barrier) and so on.

Fig. 4 shows measured and estimated model outputs that reconstruct the vehicle crash with small inaccuracies in terms of dynamic crush. The time of dynamic crush that is obtained from measurements withs models is approximately the same as the time of dynamic crush in the real-time experimental data (0,08 s). It is noticeable that when the poles of the model are close to zero, dynamic crush of the model is far from the dynamic crush of the real-time experimental data [Munyazikwiye 2014].

The fit between the two curves (reference measured output - first curve and model output - second curve) is computed in a way that 100% means a perfect fit, and 0% indicates a poor fit . This means that in our case the measured output best fits the model output in the following order: model tf1 = 99.78, model ss1 = 99.22, model P2U = 99.12, model n4s2 = 98.97, model arx441 = 94.71 and finally model arxqs = 94.70. The biggest difference in dynamic crush is observed between the measured output and the arxqs model output, the smallest between the measured output and the model output tf1, see Tab. 2.

REFERENCES

[Cichos 2006] Cichos, D. Crash Analysis Criteria Description. 2006. Available from: www.crash-network.com.

[Evin 2016] Evin, E., Tomas, M., Vyrostek, M. Laser-beam welding impact on the deformation properties of stainless steels when used for automotive applications. Acta Mechanica et Automatica, Vol. 10, no. 3, 2016, pp. 189-194. ISSN 1898-4088.

[Ljung 1994] Ljung, L., & Glad, T. (1994). Modeling of dynamic systems. Englewood Cliffs, NJ: Prentice Hall.

[Ljung 1999] Ljung, L. System identification: Theory for the user (2nd ed.). Upper Saddle River, NJ: PTR Prentice Hall.

[Marzbanrad 2011a] Marzbanrad, J. & Pahlavani, M. A system identification algorithm for vehicle lumped parameter model in crash analysis. International Journal of Modeling and Optimization, 2011, Vol. 1, No. 2, pp. 163–168.

[Marzbanrad 2011b] Marzbanrad, J. & Pahlavani, M. Calculation of vehicle lumped model parameters considering occupant deceleration in frontal crash. International Journal of Crashworthiness, 2011, Vol. 16, No. 4, pp. 439–455.

[Mathworks 2013] Mathworks. Retrieved from http://www.mathworks.com/help/toolbox/ident/

[Munyazikwiye 2013] Munyazikwiye, B.B., Karimi, H.R., Robbersmyr, K.G. Mathematical modeling and parameters estimation of car crash using eigensystem realization algorithm and curve-fitting approaches. Mathematical Problems in Engineering, 2013, 13 p. (Article ID 262196).

[Munyazikwiye 2014] Munyazikwiye, B.B. , Robbersmyr, K.G., Karimi, H.R. A state-space approach to mathematical modeling and parameters identification of vehicle frontal crash. Systems Science & Control Engineering, 2014, Vol. 2, No. 1, pp. 351-361, DOI: 10.1080/21642583.2014.883108.

[NCAP 2017] NCAP - Report number NCAP-KAR-16-014. New car assesment program. 2017. Available from: www.nrd.nhtsa.dot.gov

[Vlk 2003] Vlk, F. Construction of motor vehicles. 1st edition. Brno: 2003, 499 p. ISBN 80-238-8757-2.

CONTACTS

RNDr. Jozef Kmec, PhD.; Prof. PaedDr. Jozef Pavelka, PhD.; PaedDr. Jaroslav Šoltés, PhD.
Faculty of Humanities and Natural Sciences, Prešov University,
17 novembra 15
080 01 Prešov, Slovakia,
jozef.kmec@unipo.sk,
jozef.pavelka@unipo.sk
jaroslav.soltes@unipo.sk,