EXAMINATION OF COMPUTATIONAL PROCEDURES FROM THE POINT OF VIEW OF THEIR APPLICATIONS IN THE SIMULATION OF TORSIONAL VIBRATION IN THE MOTORCYCLE STEERING SYSTEM, WITH FREEPLAY AND FRICTION BEING TAKEN INTO ACCOUNT

DARIUSZ ŻARDECKI[1], ANDRZEJ DĘBOWSKI[2]

Military University of Technology (WAT)

Summary

The torsional vibrations that occur in the motorcycle steering system pose a significant problem for motorcyclist's safety. These vibrations are particularly conspicuous in the motorcycles operated with high dynamic loads, where the impact of freeplay and friction emerges due to mechanical wear. The vibrations generated in the motorcycle steering system in the presence of freeplay and friction have peculiar nonlinear nature typical of stick-slip processes. Due to the threshold-like phenomena reflected in the nonlinearities related to freeplay and friction, it is difficult to simulate the behaviour of such systems and comprehensive preliminary research must be first carried out for this purpose. In this paper, simulation tests of the torsional vibrations that occur in a simplified equivalent model of the steering system (torsional pendulum) have been described. A mathematical model of the system has been given, inclusive of an original method of generating external inputs. Computer software developed in the Matlab-Simulink environment has been presented. The impact of computational procedures on simulation results, including impact of the method used to implement the slick-slip model in the simulation program, impact of the type of the algorithms used to integrate differential equations, and impact of the preset numerical parameters, has been analysed.

Keywords: systems with freeplay and friction, simulation of vibrations, sensitivity analysis, computational procedures, Matlab-Simulink

[1] Military University of Technology (WAT), Faculty of Mechanical Engineering, ul. gen. Sylwestra Kaliskiego 2, 00-908 Warsaw, Poland, e-mail: dzardecki@wat.edu.pl,

[2] Military University of Technology (WAT), Faculty of Mechanical Engineering, ul. gen. Sylwestra Kaliskiego 2, 00-908 Warsaw, Poland, e-mail: adebowski@wat.edu.pl

1. Introduction

To get to know the processes that govern the motorcycle dynamics, one must thoroughly consider the issues related to torsional vibrations in the motorcycle steering system. Such vibrations result from external influences and from peculiar dynamic properties of the mechanical structure of the steering system. The nonlinearity of the torsional vibrations comes from the freeplay and friction phenomena that take place in joints between system components. The vibrations of this type are, by nature, very sensitive to changes in parameters of the mechanical structure.

Obviously, the issues related to torsional vibrations in motorcycle steering systems are perceived as important from the utilitarian point of view (safety of motorcycle riders); however, they are also attractive for scientists (nonlinear dynamics of systems functioning in the presence of freeplay and friction). Due to the difficulties encountered at the carrying out of experiments with real objects, the research on vibrations in motorcycles must be based to a significant extent on mathematical modelling and computer simulations. Meanwhile, analyses of the scientific literature dedicated to motorcycle dynamics (with important review items being e.g. [3, 4]) may lead to a conclusion that so far the issues of modelling and simulation of torsional vibrations in the motorcycle steering system are relatively seldom taken on by researchers and the freeplay and friction problems are not tackled by them at all. How can it be explained?

The modelling of dynamics of mechanical systems with freeplay and/or friction (dry, kinetic, and static) requires the use of mathematical models with "sharp" nonlinearities, including differential equations with variable structure (kinetic friction turning into static friction and vice versa, as it is in the process referred to as "stick-slip") [1, 2, 6, 7, 8]. The simulation of such systems is extremely difficult because, as it turns out, the computation results are significantly affected by the methods used to implement the model in the simulation program (including the tools used to control the variability of the structure of equations of motion), the algorithms used to integrate differential equations, including the ODE (Ordinary Differential Equation) solving procedures, and the numerical parameters of computational procedures (including the preset computational step size). Hence, not only an appropriate mathematical model has to be developed but also the methods, algorithms, and numerical parameters used in simulation programs must be thoroughly analysed before an analysis of torsional vibrations in the motorcycle steering system based on results of computer simulations is undertaken. Such a statement finds confirmation in authors' research experience and it constituted a genesis of this article.

Noteworthy is the fact that the analysis of an impact of the numerical procedures and their parameters used on the calculation results is very seldom presented in the publications where computer simulations are referred to. An example of the few publications of this kind is article [5] dealing with the testing of ODE algorithms in the simulation of dynamics of a motor car (but without the freeplay and friction having been taken into account in the vehicle model).

In this work, an attempt was made to analyse the impact of computational procedures (including the model implementation method, ODE algorithm types, and numerical parameter values) on results of the simulation of torsional vibrations in the motorcycle steering system.

2. A concept of the simulation tests

To examine the impact of the computational procedures used on the results of vibration simulations, a model was formulated that reflected the major attributes of torsional vibrations in a real steering system. The test model was simplified to the maximum extent, but with maintaining the essence of the nonlinear nature of the torsional vibrations that take place in the system in the presence of freeplay and friction. A concept of the modelling has been illustrated in Fig. 1.

Fig. 1. A concept of the equivalent model used for the testing of numerical procedures

The test model corresponds to a situation where the front motorcycle wheel does not rotate and is lifted up (there is no interaction between the wheel and the road surface) and the handlebar is fixed. The torsional vibration may be caused by the application of a variable external moment of forces or by twisting the system to move the wheel out of its angular position of equilibrium (initial conditions) and then releasing it free. This equivalent model adopted is actually a torsional pendulum where a twisted rigid inert element is coupled with a weightless elastic shaft (linear torsional elasticity) mounted with a freeplay in an external fixed rigid housing. The shaft of the inert element is placed in a bearing integrated with the external housing. The bearing acts on the twisting motion through viscous friction forces (linear damping) and dry friction forces (dry kinetic and static friction, which may cause the stick-slip phenomenon). The mathematical model that describes the torsional vibrations of the wheel (nonlinear vibrations because of the impact of freeplay and dry friction) is determined by the balance of moments of the forces of inertia, viscous friction (damping) and dry friction (kinetic and static), elastic reaction (elasticity of the column fixed with angular freeplay), and excitation (input). This model may be described in the form of a second-order differential equation with variable structure:

$$J\ddot{\alpha}(t) = \begin{cases} M_w(t) - k \cdot luz(\alpha(t),\alpha_0) - \mu \cdot tar\left(\dot{\alpha}(t), \dfrac{M_{TK0}}{\mu}\right), & gdy\ \ \dot{\alpha}(t) \neq 0 \\ luz\left(M_w(t) - k \cdot luz(\alpha(t),\alpha_0), M_{TS0}\right), & gdy\ \ \dot{\alpha}(t) = 0 \end{cases}$$

The conditions $\dot{\alpha}(t) \neq 0$ / $\dot{\alpha}(t) = 0$ have been taken from the Coulomb friction model [1]. According to the Karnopp friction model [1], the controlling of variability by adopting the conditions $|\dot{\alpha}(t)| > \varepsilon$ and $|\dot{\alpha}(t)| \leq \varepsilon$, where ε is a parameter of a "small" value, is also allowed.

Notation:
J – moment of inertia;
μ – damping coefficient (viscous friction);
M_{TK0} – moment of the dry kinetic friction forces;
M_{TS0} – maximum value of the moment of dry kinetic friction forces;
k – stiffness coefficient;
α_0 – angular freeplay parameter;
α – angle of torsion;
M_W – moment of the external input force;
t – time.

The $luz(\ldots)$ and $tar(\ldots)$representations used in the model (corresponding to freeplay and friction, respectively; cf. Fig. 2) are expressed by the following formulas:

$$luz(x,a) = x + \frac{|x-a| - |x+a|}{2} \qquad tar(x,a) = luz^{-1}(x,a)$$

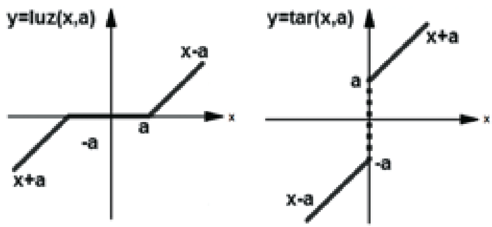


Fig. 2. Geometrical interpretation of the $luz(\ldots)$ and $tar(\ldots)$ representations

The mathematical form of the model stems from the use of the method of modelling systems with freeplay and/or friction, described in [5, 6, 7]. The $luz(\ldots)$ / $tar(\ldots)$ representations make it possible to express analytically the characteristic curves of flexibility (a stiffness curve with a "dead" zone caused by the freeplay) and kinetic friction force (a Coulomb curve, which is a superposition of a linear function and a pseudo-function $signum(\ldots)$) and, moreover, to express the stick-slip process in the neighbourhood of zero velocity.

The torsional vibrations of the pendulum may be simulated with excitation exclusively coming from non-zero initial conditions (it would be then $M_W(t) = 0$) or with external excitation $M_W(t)$ in a preset form (with zero initial conditions). The excitation coming from non-zero initial conditions leads to a situation where dry static friction develops and the motion is blocked (with the velocity being constant and equal to zero and the angular position remaining unchanged). At zero initial conditions, in turn, the excitation $M_W(t)$ appropriately applied should cause the stick-slip process to take place, with temporary stops in the motion. To expose the stick-slip process, it is advisable to apply periodical excitation with a state $M_W(t) = 0$ cyclically repeating. The use of a periodical waveform generator helps in the searching for dynamic singularities (e.g. chaos). In the face of all the above considerations, an assumption was made that at zero initial conditions, the excitation (input waveforms) $M_W(t)$ would be produced by a pre-programmed periodical waveform generator cf. Fig. 3). In the generator, a standard harmonic signal generator and a functional block implementing the *luz*(...) representation were used.

$$M_W(t) = \mathrm{luz}\left(M_{W1}\sin(2\pi ft), M_{W0}\right)$$

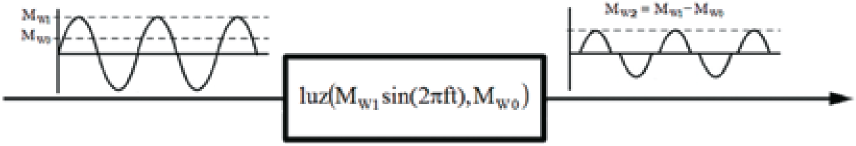


Fig. 3. Idea of generating the excitation waveform MW(t)

The effects of application of specific computational procedures in the simulation may be evaluated in various ways. For the assessment whether the state of stopping the motion is correctly simulated, the ascertaining that the angular position remains unchanged would be sufficient. The assessment of the course of the stick-slip processes at a cyclic input MW(t) being generated is more complicated. To ascertain repeatability (absence of chaos) in the simulation results, it would be sufficient to prepare appropriate graphs (e.g. "Poincare maps"). To facilitate comparisons between results of different simulation processes (e.g. for different numerical algorithms adopted to integrate the equations or for preset different equation parameters), integral indicators of sensitivity have been introduced, which are based on time histories of angular accelerations (the time histories of acceleration have been found to be more sensitive than the velocity or displacement vs. time curves).

$$s_{is} = \frac{\int_0^t \left(\ddot{\alpha}_i(t) - \ddot{\alpha}_0(t)\right)^2 dt}{\int_0^t \ddot{\alpha}_0(t)^2 dt}$$ – for algorithms with constant step size;

$$s_{iz} = \frac{\int_0^t \left(\ddot{\alpha}_i(t)^2 - \ddot{\alpha}_0(t)^2\right) dt}{\int_0^t \ddot{\alpha}_0(t)^2 dt}$$ – for algorithms with variable step size;

$\ddot{\alpha}_i(t)$ – acceleration vs. time (at discrete instants), in the simulation under examination;

$\ddot{\alpha}_0(t)$ – acceleration vs. time (at discrete instants), in the reference simulation.

The different method of calculating the indicators for the algorithms with variable step size has been dictated by possible different numbers of time points in the simulation processes.

3. Research software

The research software has been based on the Matlab-Simulink (M S) package. It makes it possible to carry out simulations of nonlinear vibrations in a torsional pendulum as well as extensive numerical research on variously defined sensitivity problems concerning both the numerical model (the issue of controlling the stick-slip process in the neighbourhood of zero velocity) and the equation integration algorithms and equation parameters. Thanks to the features of the M S package, the software having been developed enables interactive doing of calculations and this facilitates and speeds-up the research work.

The research software consists of a main program prepared as an M file in the Matlab language and a set of simulation models defined in the form of block diagrams implemented in the Simulink environment.

The main program organizes the simulation computations and the calculation of indicators, based on Simulink. From the M file level, both parameters and references to a selected model as well as a numerical procedure are defined (and modified as well). The graphs that represent simulation results are also plotted from the M file level. A block diagram of the organization of computations has been presented in Fig. 4. A listing of the main program with example data has been given in the Appendix.

Note: The assignment of the notation of parameters has been explained in a list provided at the beginning of item 4.

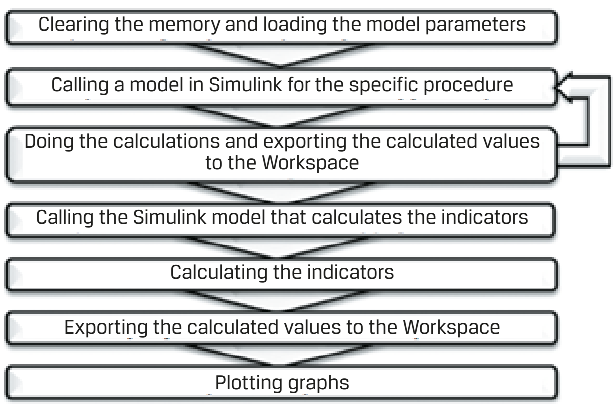


Fig. 4. Block diagram of the organization of computations

Schematic diagrams of computations implemented in the Simulink environment have been shown in Figs. 5, 6, 7, and 8.

The schematic diagram in Fig. 5 shows the general block structure of the simulation model. The macro-block includes a detailed schematic diagram of the simulation model (in various versions; examples of the most important versions have been given in Figs. 6 and 7).

The other blocks represent a virtual oscilloscope unit to monitor current traces of the quantities observed, a clock unit, generator blocks, and Workspace-type blocks, which comprise the values of the parameters preset in the M file and the time histories to be visualized from the M file level.

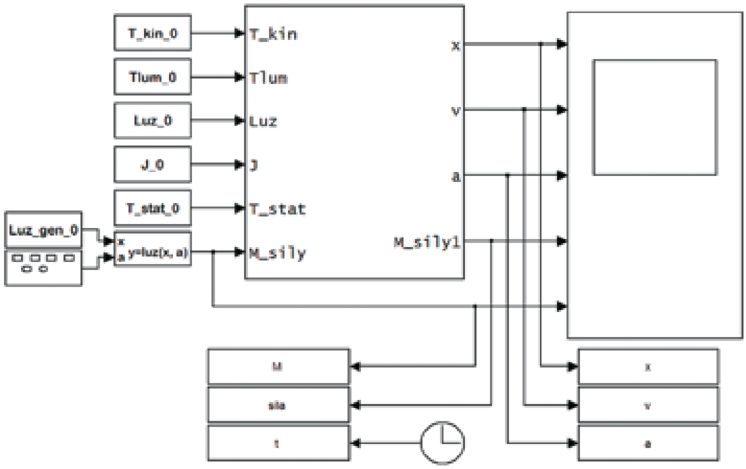


Fig. 5. Schematic diagram of the basic model structure

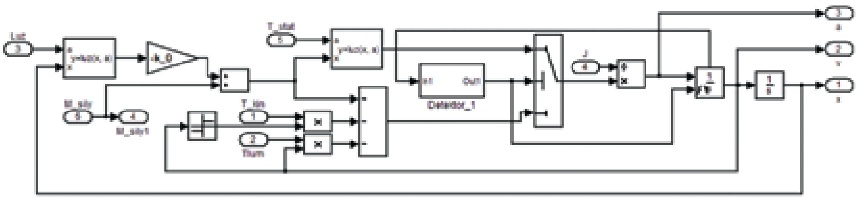


Fig. 6. Simulation model without a "hard velocity zeroing mechanism"

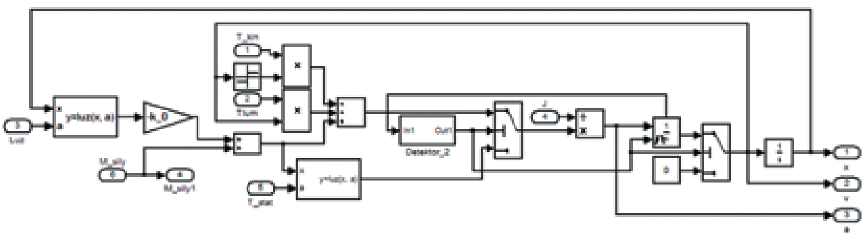
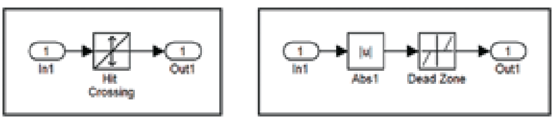


Fig. 7. Simulation model with a "hard velocity zeroing mechanism"



Detektor_1 Detektor_2

Fig. 8. Singular state detector models

The schematic diagrams shown in Figs. 6 and 7 present the essence of functioning of the computational model. In both cases, the system includes singular state detector block, model structure switch block controlled by it, and integrator block calculating the velocity values as a function of time and provided with an additional resetting input and velocity state output. For the whole reset signal duration time, the integrator remains in its initial state. The reset signal is generated by the detector block in a closed loop system, based on the velocity state output signal. This signal becomes available earlier than the standard signal at the integrator output, thanks to which an algebraic loop in the computation process is avoided. The control condition ($\dot{\alpha}(t) \neq 0$ / $\dot{\alpha}(t) = 0$ or $|\dot{\alpha}(t)| > \varepsilon$ and $|\dot{\alpha}(t)| \leq \varepsilon$) may be defined in the detector in various ways, which are equivalent to each other in terms of their effects (Fig. 8). When a detector based on a Hit Crossing block is used, the parameter ε is set through the offset parameter of the block.

A distinguishing feature of the schematic diagram shown in Fig. 7 is an additionally applied "mechanism of hard zeroing" of velocity when the condition of stopping the motion begins to be fulfilled. Thanks to this, stability of the numerical process is ensured in the situations where the computational model without such a mechanism fails to function properly.

The M S package offers its user a number of algorithms intended for numerical integration of differential equations [9]. They include algorithms with constant step size (ODE1, ODE2, ODE3, ODE4, ODE5, ODE14x) and with variable step size (ODE45, ODE23, ODE113, ODE15s, ODE23s, ODE23t, ODE23tb). An algorithm called from the M file is implemented in Simulink blocks of the Integrator type, provided in schematic diagrams of the simulation models.

Particularly noteworthy is the algorithm ODE1 implementing the simplest first-order Euler's procedure, present also as a component in other algorithms that are more complicated, e.g. ODE4 or ODE45, and based on the Runge-Kutta 4th order method. For this reason, the ODE1 algorithm is taken in comparative investigations as a reference.

4. Simulation tests

Simulation tests of torsional vibrations, carried out with the torsional pendulum presented above being taken as a basis, were oriented at testing various model implementation methods, various equation integration algorithms, and variously assumed numerical parameters. The tests were repeated for many variants of data sets concerning the physical parameters of the pendulum model. In this paper, only example results of simulation tests have been given because of editorial limitations. They were obtained for parameters adopted for the model (the test parameters specified in SI units are by no means data of a real steering system; nevertheless, they may be treated as parameters of a calibrated model):

J_0 = 0.5 – moment of inertia;

Tlum_0=0.5 – damping coefficient (for viscous friction);

T_kin = 0.20 – moment of dry kinetic friction;

T_stat = 0.25 – maximum moment of static friction;

k_0=100 – stiffness coefficient;

Luz_0= 0.01 – angular freeplay parameter;
Luz_gen_0=0.2 – "freeplay" parameter in the generator of external input.

The findings made in result of the whole research work have been formulated in the final conclusions.

Examining the impact of model implementation method on simulation results

The tests were carried out with using various simulation model diagrams. The primary purpose of the tests was to verify the correctness of model functioning at various implementations of the block diagram of the model, to observe the nature of changes in the test results depending on which parameter was modified, and to select the parameters the values of which would be subsequently used as references in the comparative model.

Initially, vibrations were induced as an effect of a non-zero initial angle of torsion (non-zero initial condition), without no other external excitation being applied. The simulation model without a "hard velocity zeroing mechanism" behaved improperly in many cases, i.e. in spite of a singular state being detected, the system motion was not completely blocked (a numerical zero-velocity noise remained present in the system, resulting after integration in a position drift). Such shortcomings did not occur when a simulation model with a "hard velocity zeroing mechanism" was used (cf. Fig. 9a). This model proved also to be reliable when a cyclic external input was applied (cf. Fig. 9b). For these reasons, a decision was made to carry out all the subsequent tests only with the use of the simulation model presented in Fig. 7.

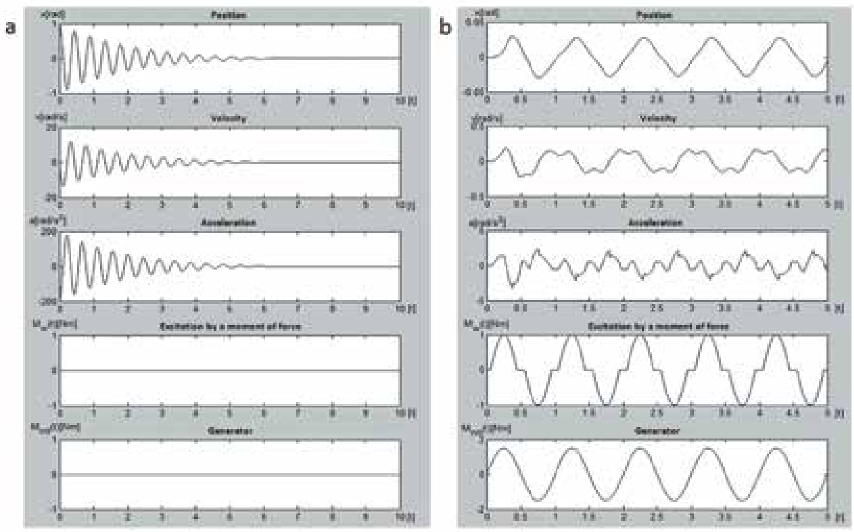


Fig. 9. Example simulation results obtained at preliminary tests:
a – vibrations exclusively induced by a non-zero initial condition; b – vibrations induced by an external input.
The results were obtained at the procedure ODE1 with constant step size.
Numerical parameters: $\Delta t = 0.0001$, $\varepsilon = 0.0001$

Examining the impact of ODE algorithm type on simulation results

These tests were carried out to ascertain the degree of discrepancy between simulation results depending on the ODE numerical procedure used. The graphs shown in Figs. 10, 11, 12 and 13 present graphical comparisons between simulation results obtained at both vibration excitation types with the use of all the ODE algorithms available in the Simulink package. In consideration of a large number of the ODE algorithms tested, the graphs have been divided into two groups according to procedures with constant and variable step size. The following numerical parameters were used at the calculations: Δt = 0.0001 (for the procedures with constant step size and as the starting step size for the procedures where the step size varied), ε = 0.0001.

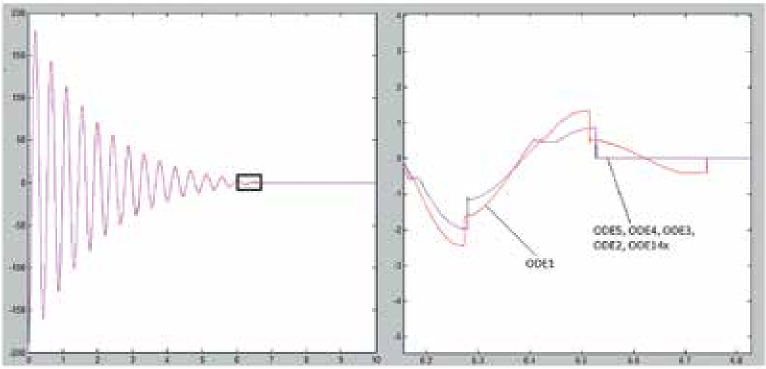


Fig. 10. Comparison of time history curves for the excitation exclusively coming from non-zero initial conditions, for the procedures with constant step size

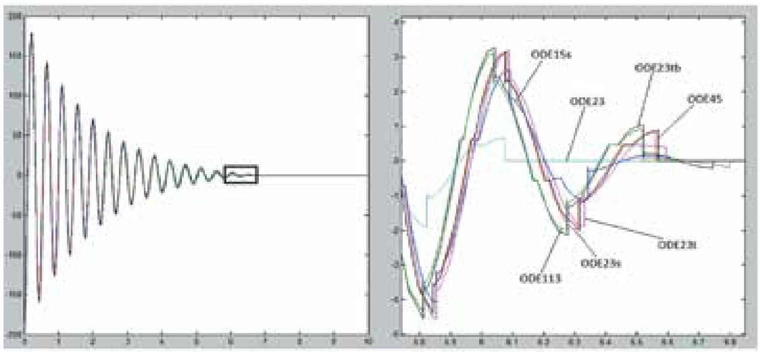


Fig. 11. Comparison of time history curves for the excitation exclusively coming from non-zero initial conditions, for the procedures with variable step size

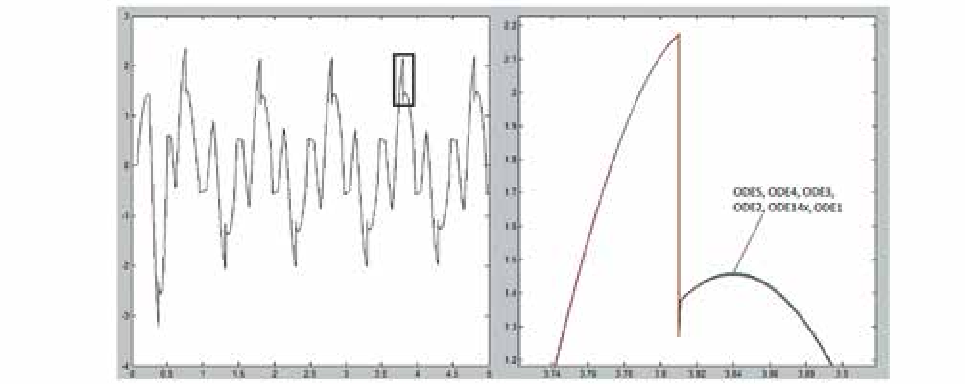


Fig. 12. Comparison of time history curves for the external excitation, for the procedures with constant step size

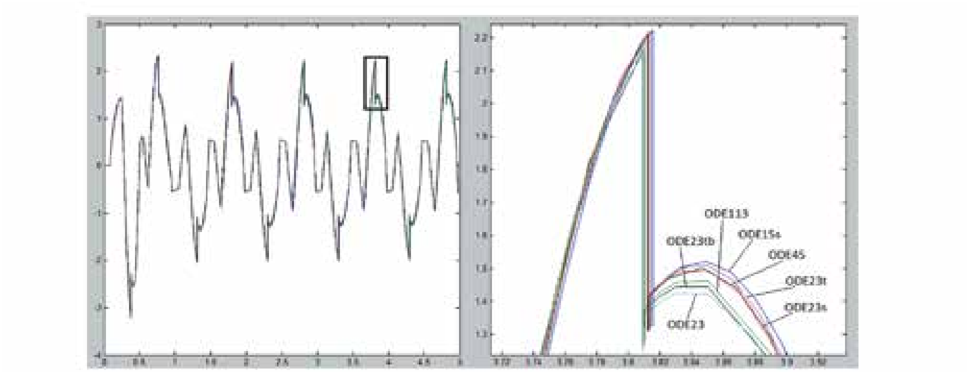


Fig. 13. Comparison of time history curves for the external excitation, for the procedures with variable step size

As it can be seen, the differences between values of the simulation results obtained with the use of the procedures with constant step size were smaller (the graphs almost coincide with each other) than they were in the case of the procedures with variable step size. These differences are particularly conspicuous for the excitation exclusively coming from non-zero initial conditions.

Table 1. Summarized values of the integral indicator of sensitivity for specific ODE procedures

Procedures with constant step size					
ODE5	ODE4	ODE3	ODE2	ODE1	ODE14x
0.000356319	0.000310396	0.000310582	0.000232692	0.000000000	0.000464708

Procedures with variable step size						
ODE45	ODE23	ODE113	ODE15s	ODE23s	ODE23t	ODE23tb
-0.001124013	-0.001873847	-0.000382024	0.005858206	-0.007509621	0.005734172	0.009594022

Examining the impact of integration step size Δt on simulation results

One of the most important parameters in simulation programs is the integration step size Δt. In the case of the procedures with constant step size, it is defined as the computational step size maintained during the whole simulation process; for the procedures with variable step size, it has the meaning of the starting step size. The selection of an optimum value of the step size is essential for the computation accuracy and completion time. Inappropriate Δt may even result in numerical chaos in the case of external periodical excitation.

The time history curves shown in Fig. 14 represent example simulations carried out at ε = 0.0001 for successively changed step size Δt. For the sake of clarity of the results obtained, the presentation has been limited to the algorithms that are most typical for the M S software, i.e. ODE1, ODE4, and ODE45.

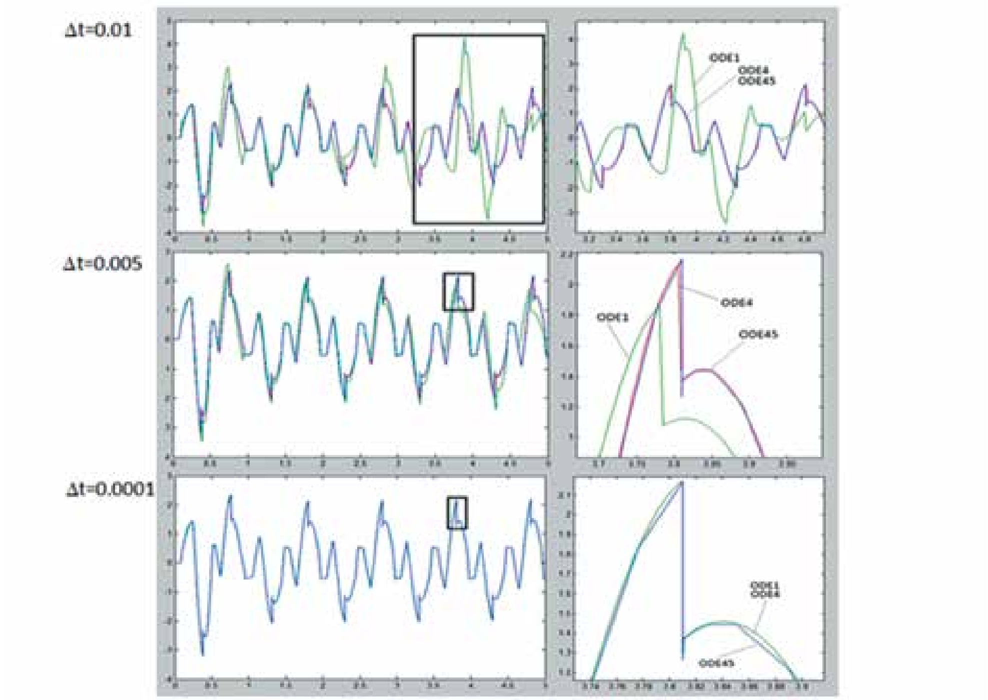


Fig. 14. Comparison of time history curves for procedures ODE1, ODE4, and ODE45 at various Δt values

As it can be seen in the graphs, the ODE1 algorithm is particularly sensitive to changes in the integration step size. This fact is well known from the theory. On the other hand, if the integration step size is sufficiently small and properly selected then the results obtained with the use of this algorithm do not significantly differ from those received from other procedures. This is clearly confirmed by the sensitivity indicator values brought together in Table 2.

Table 2. Summarized values of the integral indicator of sensitivity for various integration steps

Procedures with constant step size						
Δt	ODE5	ODE4	ODE3	ODE2	ODE1	ODE14x
10^{-2}	0.82610615	0.86805045	0.80486410	0.95624965	0.00000000	1.02988909
$5*10^{-3}$	0.11104146	0.10765974	0.12314039	0.10384573	0.00000000	0.18253272
10-3	0.00489426	0.00443881	0.00467866	0.00372494	0.00000000	0.00846028
10^{-4}	0.00035632	0.00031040	0.00031058	0.00023269	0.00000000	0.00046471
10^{-5}	0.00002559	0.00002110	0.00002408	0.00002258	0.00000000	0.00018296
10^{-6}	0.00000225	0.00000240	0.00000270	0.00000270	0.00000000	0.00001753

Examining the impact of parameter ε on simulation results

Parameter ε delimits the angular velocity range that governs the switching over between the kinetic friction and static friction models. The selection of its value will have an effect on the instant of switching over from one model structure to another and on the shape of the curve representing the time history of the quantity simulated. Thus, this parameter will significantly influence the simulated stick-slip process in the case of external excitation by periodically changing moment of forces.

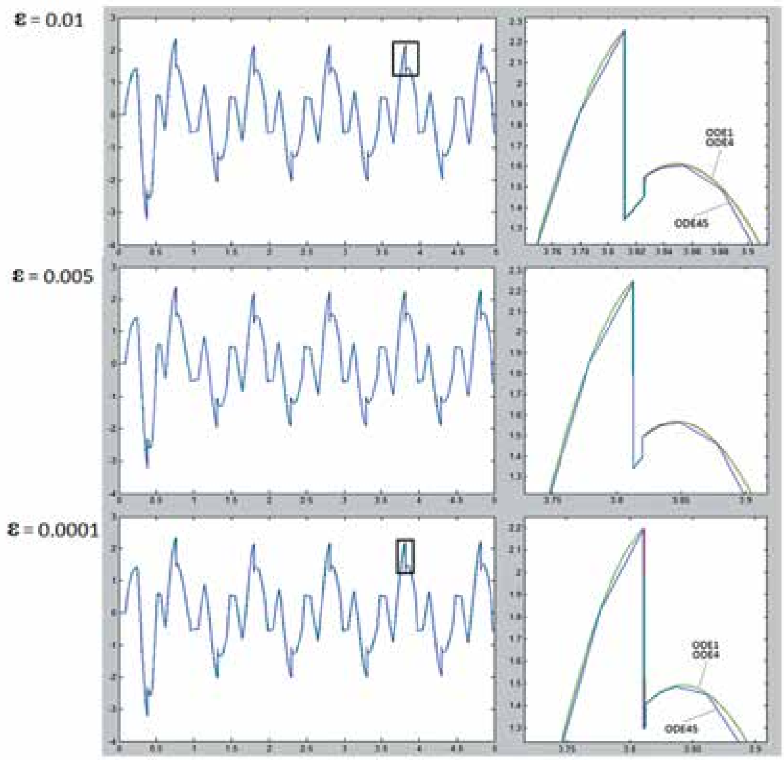


Fig. 15. Comparison of time history curves for procedures ODE1, ODE4, and ODE45 at various ε values

Representative time history curves for simulations with various integration procedures at Δt = 0.0001, where parameter ε was modified, have been shown below (Fig. 15). As it can be seen, the higher value is given to parameter ε, the earlier the equation structure corresponding to static friction is switched on and for the longer time this structure remains active. Simultaneously, the impact of the type of the integration algorithm used decreases. Obviously, excessive length of this time interval would mean certain divergence from the physical interaction between elements of the friction pair; nevertheless, the ε value on a level of 0.0001 makes it possible to obtain a sufficiently good representation of the friction process (the Karnopp model).

Table 3a. Summarized values of the integral indicator of sensitivity for various ε values

Procedures with constant step size						
ε	ODE5	ODE4	ODE3	ODE2	ODE1	ODE14x
10^{-2}	0.02497189	0.02512368	0.02507356	0.02501780	0.02590390	0.02476825
$5*10^{-3}$	0.01352245	0.01360921	0.01365632	0.01380132	0.01407273	0.01299544
10^{-3}	0.00283269	0.00286885	0.00293402	0.00278653	0.00274506	0.00259051
$5*10^{-4}$	0.00146052	0.00143889	0.00147724	0.00151607	0.00137171	0.00149539
10^{-4}	0.00035632	0.00031040	0.00031058	0.00023269	0.00000000	0.00046471
10^{-5}	0.00031307	0.00031350	0.00033923	0.00036767	0.00041445	0.00237397
10^{-7}	0.00057838	0.00045448	0.00050659	0.00051853	0.00066416	0.00049414
10^{-11}	0.00052219	0.00046666	0.00050757	0.00053166	0.00066416	0.00038598

Table 3b. Summarized values of the integral indicator of sensitivity for various ε values

Procedures with variable step size							
ε	ODE45	ODE23	ODE113	ODE15s	ODE23s	ODE23t	ODE23tb
10^{-2}	0.00780631	0.00471071	0.00661056	0.00610657	-0.00130526	0.01036063	0.01428781
$5*10^{-3}$	0.00727367	0.00538441	0.00570064	0.01106889	-0.00062328	0.01254941	0.01557724
10^{-3}	0.00188459	0.00061212	0.00252601	0.01018416	-0.00475035	0.00915447	0.01291723
$5*10^{-4}$	0.00055198	-0.00048191	0.00139228	0.00887768	-0.00600282	0.00809743	0.01127922
10^{-4}	-0.00112401	-0.00187385	-0.00038202	0.00585821	-0.00750962	0.00573417	0.00959402
10^{-5}	-0.00162037	-0.00285782	-0.00255050	0.00091112	-0.00870358	0.00431328	0.00873918
10^{-7}	-0.00245297	-0.00305358	-0.00039953	0.00392048	-0.00900668	0.00447247	0.00808097
10^{-11}	-0.00253423	-0.00290871	0.00116674	0.00457220	-0.00902241	0.00421940	0.00813836

A reduction of the ε values causes at the same time a decline in the indicator value; however, the changes become insignificant for ε values below 0.0001. Noteworthy is the considerable convergence of results of the procedures with constant step size, where the values of the results only slightly differ from each other, while this cannot be said about the procedures with variable step size.

5. Conclusions and final remarks

The simulation experiments carried out, some fragments of which have been presented here, gave grounds for formulating a number of practical conclusions and findings as regards the simulation of vibrations that may occur in the motorcycle steering system, with freeplay and friction in the system being taken into account. The conclusions and final remarks have been given below.

- The simulation of torsional vibrations in a system with freeplay and friction is generally quite difficult, because the computation results are considerably affected by the methods used to implement a model in the simulation program, the differential equation integration algorithms employed, and the numerical parameter values adopted in the computational procedures.
- In the numerical model, it is advisable to introduce a "mechanism of hard zeroing" of the velocity calculated, applicable to the states that cause the system motion to be blocked by dry friction. The absence of such a "mechanism" often results in a drift shown in the time history of the angular position of the torsional pendulum.
- It has been found legitimate to use the simple ODE1 algorithm with constant step size if the numerical parameters Δt and ε are properly selected. For the pendulum model data set as presented in this paper, satisfactory results were obtained at $\Delta t = 0.0001$ and $\varepsilon = 0.0001$. In comparison with more complicated procedures, the ODE1 algorithm turns out to be particularly useful at the computations where the freeplay and dry friction parameter values are high. The ODE1 algorithm is considered superior to other procedures because the more complex algorithms give results with the averaging of partial computations and this causes the stick-slip processes to be smoothed and, simultaneously, introduces effects of numerical chaos, manifesting themselves in randomness of the time history curves recorded at cyclic inputs. This particularly applies to complex procedures with variable step size, where relatively short computation time remains their only good point in such a case.
- The simulation results are highly sensitive to changes in the numerical parameters Δt and ε; this has been observed for all the algorithms used. Based on tests carried out for other sets of physical model parameters, a statement should be made that the correctness of selection of numerical parameters must in every case be confirmed by comparative tests.

References

[1] ARMSTRONG-HELOUVRY B., DUPONT P., CANUDAS DE WIT C.: *A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction.* Vol. 30, No. 7, Automatica, 1994, s. 1083–1138.

[2] BROGLIATO B., DAM A., PAOLI L., GENOT F., ABADIE M.: *Numerical simulation of finite dimensional multibody nonsmooth mechanical systems.* Applied Mechanical Review, ASME,2002, Vol. 55, No. 2, s. 107–150.

[3] COSSALTER V.: *Motorcycle dynamics.* UK, 2006.

[4] PACEJKA H. B.: *Tyre and vehicle dynamics.* Oxford, Elsevier, 2012.

[5] SZCZOTKA M., TENGLER S., WOJCIECH S.: *Numerical effectiveness of models and methods of integration of the equations of motion of a car.* Differential Equations and Nonlinear Mechanics. Art. ID 49157, 2007.

[6] ŻARDECKI D.: *Piecewise Linear luz(…) and tar(…) Projections. Part 1 – Theoretical Background.* Journal of Theoretical and Applied Mechanics, 2006, Vol. 44, No. 1, s. 163–184.

[7] ŻARDECKI D.: *Piecewise Linear luz(…) and tar(…) Projections. Part 2 – Application in Modeling of Dynamic Systems with Freeplay and Friction.* Journal of Theoretical and Applied Mechanics, 2006,Vol. 44, No. 1, s. 185–202.

[8] ŻARDECKI D.: *Modelowanie luzu i tarcia oparte na odwzorowaniach luz(…) i tar(…) – podstawy teoretyczne i zastosowanie w symulacji drgań nieliniowych w układach kierowniczych samochodów.* (disertation) Warszawa, Wojskowa Akademia Techniczna, 2007.

[9] Matlab-Simulink product documentation, http://www.mathworks.com/

Appendix: Listing of the M file

```
clear all;
clc;

%% Parametrymodelu                                              (Model parameters)
War_pocz=0;
War_pocz_0=War_pocz;
Ampl=1.5;
Ampl_0=Ampl;
Cz=1;
Cz_0=Cz;
dt=0.0001;
t_stop=5;

%% Parametry modelu odniesienia                                 (Reference model parameters)
J=0.5;
Tlum=0.5;
T_kin=0.20;
T_stat=0.25;
k=100;
Luz=0.01;
Luz_gen=0.5;
e=0.0001;

%% parametry modeli dla wszystkich procedur                     (Model parameters for all the
                                                                procedures)
J_0=0.5;
Tlum_0=0.5;
T_kin_0=0.2;
T_stat_0=0.25;
k_0=100;
Luz_0=0.01;
Luz_gen_0=0.5;
e_0=0.0001;

%% procedura numeryczna dla każdego modelu                      (Numerical procedure for every
                                                                model)
sim('Model0_ode1_odniesienie');  %Wywołanie modelu simulinkowego (Call for the Simulink model)
set_param('Model0_ode1_odniesienie', 'StopTime', 't_stop');
sim('Model0_ode5');  %Wywołanie modelu simulinkowego            (Call for the Simulink model)
.
.
sim('Model0_ode23tb');  %Wywołanie modelu simulinkowego         (Call for the Simulink model)
set_param('Model0_ode23tb', 'StopTime', 't_stop');

%% wartości wejściowe przyśpieszenia                            (Acceleration input values)
nominalna_a0=a; % przebieg odniesienia ode1                     (Reference time history curve)
porownywana_a1=a0_2; %ode5
porownywana_a2=a0_3; %ode4
porownywana_a3=a0_4; %ode3
porownywana_a4=a0_5; %ode2
porownywana_a5=a0_6; %ode1
porownywana_a6=a0_7; %ode14x

%% Model obliczania wskaźnika                                   (Indicator computation model)
sim('Wskaznik_przyśpieszenia');                                 (Acceleration indicator)
set_param('Wskaznik_przyśpieszenia', 'StopTime', 't_stop');     (Acceleration indicator)
```

```
%% zapis do excel                                                   (Writing in Excel)
xlswrite('Wykresy', W_a1, 'sheet', 'B3');                           ('Wykresy' = 'Graphs")
xlswrite('Wykresy', W_a2, 'sheet', 'C3');
xlswrite('Wykresy', W_a3, 'sheet', 'D3');
xlswrite('Wykresy', W_a4, 'sheet', 'E3');
xlswrite('Wykresy', W_a5, 'sheet', 'F3');
xlswrite('Wykresy', W_a6, 'sheet', 'G3');
xlswrite('Wykresy', C_ode45, 'sheet', 'H3');
xlswrite('Wykresy', C_ode23, 'sheet', 'I3');


xlswrite('Wykresy', C_ode113, 'sheet', 'J3');
xlswrite('Wykresy', C_ode15s, 'sheet', 'K3');
xlswrite('Wykresy', C_ode23s, 'sheet', 'L3');
xlswrite('Wykresy', C_ode23t, 'sheet', 'M3');
xlswrite('Wykresy', C_ode23tb, 'sheet', 'N3');

%% Wykresy 2D                                                       (2D graphs)
figure('name', 'ODE1, ODE4, Ode45', 'Position', [70 5 690 440]);
plot(t0_3,a0_3, 'r',t0_6,a0_6, 'g',t0_9,a0_9, 'b');
```