KORNEL WARWAS[*]
SZYMON TENGLER[**]

DYNAMIC OPTIMISATION OF MULTIBODY SYSTEM USING MULTI-THREAD CALCULATIONS

A b s t r a c t

The paper presents dynamic optimization methods used to calculate the optimal braking torques applied to wheels of an articulated vehicle in the lane following/changing maneuver in order to prevent a vehicle rollover. In the case of unforeseen obstacles, the nominal trajectory of the articulated vehicle has to be modified, in order to avoid collisions. Computing the objective function requires an integration of the equation of motions of the vehicle in each optimization step. Since it is rather time-consuming, a modification of the classical gradient method - Variable Metric Method (VMM) was proposed by implementing parallel computing on many cores of computing unit processors. Results of optimization calculations providing stable motion of a vehicle while performing a maneuver and a description and results of parallel computing are presented in the paper.

Keywords: parallel computing, multithreading, dynamic optimization, articulated vehicle

1. Introduction

Rollover accidents of articulated vehicles cause a greater damage and injury than other accidents. The relatively low roll stability of trucks promotes rollovers and contributes to lots of truck accidents [1, 2]. Research by the National Highway Traffic Safety Administration in the United States shows that the rollover accidents are the second most dangerous form of accidents in the Unites States, after head-on collisions [3]. Anti-lock Braking Systems (ABS), Electronic Braking Systems (EBS) and Electronic Stability Programs (ESP) all help in preventing vehicle rollovers, as they can automatically adjust the braking torques for each wheel what can provide a driver

[*] PhD, Kornel Warwas, Department of Computer Science and Engineering, University of Bielsko-Biała.

[**] PhD, Szymon Tengler, Department of Mechanics, University of Bielsko-Biała.

with greater control [2, 4, 5, 6]. Lots of companies introduce their own solutions which are to improve vehicle stability. An example can be Wabco which within the Smart Track program implements Electronic Stability Control (ESC) and Roll Stability Control (RSC) systems. ESC can assist the driver in reducing the risk of vehicle instability while in a slippery curve or taking an evasive action, reducing the potential for jackknifing and drift-out conditions through a selection of braking of the tractor and an application of the trailer brakes which provide both Roll Stability Control (RSC) functionality and added directional stability control. The system checks and updates the lateral acceleration of the tractor and compares it to a critical threshold at which a rollover can occur. When the critical threshold is met, RSC intervenes by reducing an engine torque and engaging the engine retarder while automatically applying drive axle and trailer brakes. Many other subsystems such as active front steering (AFS), active braking (AB), and active suspension (AS) control for rollover prevention have been carried out based on a full vehicle model are also mentioned in [7].

A design of virtual computational models enables the number of experimental road tests of real vehicles to be decreased [5, 8]. In the paper, the method of maintaining stability during an untripped rollover maneuver of an articulated vehicle is formulated. The method is based on the control of braking torques in the case of losing the stability. Braking torques patterns, which have to be applied to each wheel of the vehicle, are obtained by solving an optimization task. Since the optimization task is time-consuming, parallel computing was implemented on lots of cores of the computing unit processors. Parallel computing is a subject of many investigations and it is more and more frequently used in engineering computing [9, 10, 11]. In paper [11] possibilities which provide parallel computing in the area of the multi-body systems are presented, and the authors point out to reducing computation time from one to two orders of magnitude. In the dynamic optimization field computational intelligence algorithms, such as genetic and evolutionary algorithms which can be implemented in a simple way by use of multi-core calculations, are the most rapidly developing methods [12, 13]. For an example, the authors of paper [13] used the multi-core calculations in the Dual Population Genetic Algorithm (DPGA) method, and in [4] parallel computing and distributed systems were used to optimize satellite vibrations while performing a maneuver of laying down solar panels. The known algorithm named Island Model Genetic Algorithm was used here. Similar approaches applied in order to speed up the optimization process by parallelization of the optimization algorithms have been presented in [15]. A flaw of distributed computing is a great implementation input connected with low-level operations of data synchronization between the computing cluster nodes. As it is shown in paper [16] the multi-core calculations are free from those flaws, and additionally, an increase in efficiency can be significant. The multi-core calculations can be also applied to classical gradient and non-gradient optimization methods. A way of modifying the Variable Metric Method (VMM) enabling to determine the most time-consuming computing operations in a parallel way is presented in this paper.

2. Mathematical model

A spatial model of an articulated vehicle has been formulated as a system of rigid bodies: a tractor, a fifth wheel and a semitrailer, forming an open kinematic chain (fig. 1).

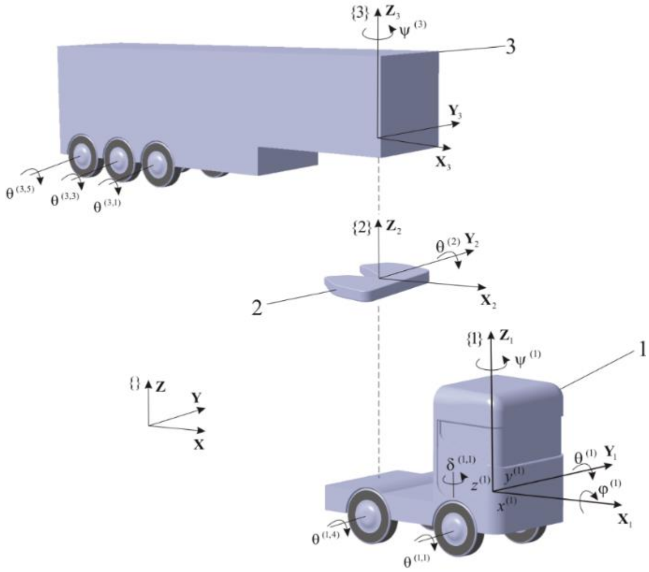

Fig. 1. A model of the articulated vehicle
1) the tractor, 2) the fifth wheel, 3) the semi-trailer

It is assumed that the tractor motion is described by means of six generalized coordinates, the fifth wheel has one degree of freedom (a pitch angle) in relation to the trailer, the semitrailer has one degree of freedom (an inclination angle) with respect to the fifth wheel. The wheels are connected with the trailer and semi-trailer and each has one degree of freedom. The tractor consists of four wheels and the semitrailer has six wheels. Suspension stiffness has been reduced to a contact point of a tire with a road. Additionally, the model consists of generalized coordinates which are front wheels steering angles of the trailer. Generalized coordinates vector of the articulated vehicle can be written in the following form:

$$\mathbf{q} = \begin{bmatrix} \tilde{\mathbf{q}}_T^{(1)} & \tilde{\mathbf{q}}_F^{(2)} & \tilde{\mathbf{q}}_S^{(3)} & \tilde{\mathbf{q}}_{TS}^{(1)} & \tilde{\mathbf{q}}_{TW}^{(1)} & \tilde{\mathbf{q}}_{SW}^{(3)} \end{bmatrix}^T \tag{1}$$

where:

$\tilde{\mathbf{q}}_T^{(1)} = \begin{bmatrix} x^{(1)} & y^{(1)} & z^{(1)} & \psi^{(1)} & \theta^{(1)} & \varphi^{(1)} \end{bmatrix}^T$ – the generalized coordinates of the tractor,

$\tilde{\mathbf{q}}_{TS}^{(1)} = \begin{bmatrix} \delta^{(1,1)} & \delta^{(1,2)} \end{bmatrix}^T$ – the generalized coordinates of the tractor suspension,

$\tilde{\mathbf{q}}_{TW}^{(1)} = \begin{bmatrix} \theta^{(1,1)} & \theta^{(1,2)} & \theta^{(1,3)} & \theta^{(1,4)} \end{bmatrix}^T$ -the generalized coordinates of the tractor wheels,

$\tilde{\mathbf{q}}_F^{(2)} = \left[\theta^{(2)}\right]^T$ - the generalized coordinates of the fifth wheel,

$\tilde{\mathbf{q}}_S^{(3)} = \left[\psi^{(3)}\right]^T$ - the generalized coordinates of the semi-trailer,

$\tilde{\mathbf{q}}_{SW}^{(3)} = \left[\theta^{(3,1)} \quad \theta^{(3,2)} \quad \theta^{(3,3)} \quad \theta^{(3,4)} \quad \theta^{(3,5)} \quad \theta^{(3,6)}\right]^T$ - the generalized coordinates of the semi-trailer wheels,

$x^{(i)}$, $y^{(i)}$, $z^{(i)}$ - the mass center coordinates of the i-th body,

$\psi^{(i)}$, $\theta^{(i)}$, $\varphi^{(i)}$ - the rotation angles of the i-th body,

$\delta^{(i)}$ - the front wheels steering angle of the vehicle.

The equations of vehicle motion have been formulated using Lagrange formalism and homogenous transformations [17]. It can be written in the general form:

$$\begin{aligned} \mathbf{A}\ddot{\mathbf{q}} - \mathbf{\Phi}\mathbf{R} &= \mathbf{f} \\ \mathbf{\Phi}^T\ddot{\mathbf{q}} &= \mathbf{W} \end{aligned} \qquad (2)$$

where:

$$\mathbf{A} = \mathbf{A}(t,\mathbf{q}) = \begin{bmatrix} \mathbf{A}_{T,T}^{(1)} + \mathbf{A}_{T,T}^{(2)} + \mathbf{A}_{T,T}^{(3)} & \mathbf{A}_{T,F}^{(2)} + \mathbf{A}_{T,F}^{(3)} & \mathbf{A}_{T,S}^{(3)} & \mathbf{A}_{T,TS}^{(1)} & \mathbf{A}_{T,TW}^{(1)} & \mathbf{A}_{T,SW}^{(3)} \\ \mathbf{A}_{F,T}^{(2)} + \mathbf{A}_{F,T}^{(3)} & \mathbf{A}_{F,F}^{(2)} + \mathbf{A}_{F,F}^{(3)} & \mathbf{A}_{F,S}^{(3)} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{F,SW}^{(3)} \\ \mathbf{A}_{S,T}^{(3)} & \mathbf{A}_{S,F}^{(3)} & \mathbf{A}_{S,S}^{(3)} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{S,SW}^{(3)} \\ \mathbf{A}_{TS,T}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{TS,TS}^{(1)} & \mathbf{A}_{TS,TW}^{(1)} & \mathbf{0} \\ \mathbf{A}_{TW,T}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{TW,TS}^{(1)} & \mathbf{A}_{TW,TW}^{(1)} & \mathbf{0} \\ \mathbf{A}_{SW,T}^{(3)} & \mathbf{A}_{SW,F}^{(3)} & \mathbf{A}_{SW,S}^{(3)} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{SW,SW}^{(3)} \end{bmatrix}$$

- the mass matrix,

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ 1 & 0 \\ 0 & 1 \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$ -the constraints matrix,

$$\mathbf{f} = \mathbf{f}\left(t, \mathbf{q}, \ddot{\mathbf{q}}, \mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(i)}, \ldots, \mathbf{M}^{(n_W)}\right) = \begin{bmatrix} \mathbf{f}_T^{(1)} + \mathbf{f}_T^{(2)} + \mathbf{f}_T^{(3)} \\ \mathbf{f}_F^{(2)} + \mathbf{f}_F^{(3)} \\ \mathbf{f}_S^{(3)} \\ \mathbf{f}_{TS}^{(1)} \\ \mathbf{f}_{TW}^{(1)} \\ \mathbf{f}_{SW}^{(3)} \end{bmatrix}$$ - a vector of external forces,

$\mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$ - a vector of unknown constraint reactions,

$\mathbf{w} = \begin{bmatrix} \ddot{\delta}^{(1,1)} \\ \ddot{\delta}^{(1,2)} \end{bmatrix}$ - a vector of right sides of constraint equations.

$\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ - displacement, velocity and acceleration vectors,

$\mathbf{M}^{(i)}$ - a vector of discrete values of braking torques acting on the *i*-th wheel,

n_w - a number of wheels,

The details of the procedure which lead to form equation (2) with a description of elements in matrices $\mathbf{A}$, $\mathbf{A}_{i,j}^{(k)}$ and vectors $\mathbf{f}$, $\mathbf{f}_i^{(k)}$ are presented in [8].

3. Formulation of control problem

The most dangerous road maneuver for articulated vehicles is a rollover. This situation happens mostly during the unforeseen lane-change maneuver [5]. Such maneuvers are performed when the preplanned vehicle trajectory would collide with an obstacle. When the obstacle is detected, the trajectory is translated to other traffic lane in order to avoid collisions with the obstacle as shown in fig. 2.

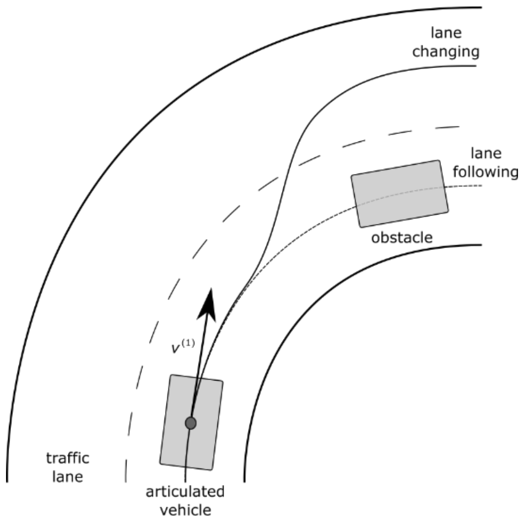

Fig. 2. Lane changing and lane following maneuver during cornering

The stability of the articulated vehicle can be restored by an appropriate control of the braking torques applied to each wheel of the vehicle. Let us consider the vector of braking torque discrete values $\mathbf{M}^{(i)}$ of *i*-th wheel. A continuous function $M^{(i)}(t)$ will be obtained using spline functions of the 3rd order. The vector of the decisive variables contains of discrete values of the braking torques of the wheels and it can be written in the form:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}^{(1)} & \dots & \mathbf{M}^{(i)} & \dots & \mathbf{M}^{(n_w)} \end{bmatrix}^T \tag{3}$$

where:

$\mathbf{M} = (M_j)_{j=1,\dots,m}$,

m - a number of decisive variables,

$\mathbf{M}^{(i)} = \left(M_k^{(i)}\right)_{k=1,\dots,n}$,

n - a number of discrete values of the braking moment.

Additionally, constraints determining the minimum and maximum values of the braking moments were taken into consideration :

$$M_{\min} \le M_k^{(i)} \le M_{\max} \tag{4}$$

where $M_{\min}$, $M_{\max}$ - the minimum and maximum value of the braking moment in a discrete time moment. In the considered optimisation problem, inequality constraints can be written in the vector form as follows:

$$\begin{bmatrix} M_{\min} - M_1 & \dots & M_{\min} - M_i & \dots & M_{\min} - M_m \end{bmatrix}^T \le \mathbf{0} \tag{5}$$

$$\begin{bmatrix} M_1 - M_{\max} & \dots & M_i - M_{\max} & \dots & M_m - M_{\max} \end{bmatrix}^T \le \mathbf{0} \tag{6}$$

and a general form of the inequality constraints can be written as follows [18]:

$$g_i(\mathbf{M}) \le 0 \tag{7}$$

where: $i = 1,\dots,n_g$,

n_g - a number of the inequality constraints.

The constraints were taken into account in the optimization tasks by an external penalty function [18], which was added to the base objective function:

$$\gamma_i(\mathbf{M}) = \begin{cases} 0 & \text{dla} \quad g_i(\mathbf{M}) \le 0 \\ C_{1,i} \exp\left(C_{2,i}\, g_i(\mathbf{M})\right) & \text{dla} \quad g_i(\mathbf{M}) > 0 \end{cases} \tag{8}$$

where $C_{1,i}$, $C_{2,i}$ are weights selected empirically.

The stability conditions can be assured by solving a dynamic optimization problem in the general form [18, 19]:

$$\Omega(\mathbf{M}, \mathbf{q}, \dot{\mathbf{q}}) \to \min \tag{9}$$

The calculations of the objective function (9) require an integration of model equations (2). In the presented problem braking torques calculated for a fixed initial vehicle velocity and front wheels steering angle have to fulfil the following conditions:

- the articulated vehicle cannot lose stability during a manoeuvre,
- the total velocity loss has to be as small as possible.

The above assumptions are taken into account in the objective function which can be written in the following form:

$$\Omega(\mathbf{M},\mathbf{q},\dot{\mathbf{q}}) = \frac{1}{t_e}\left(C_1 \int_0^{t_e} \left(\varphi^{(1)}\right)^2 dt + C_2\left(v_0 - v_e^{(1)}\right)\right) + \sum_{i=1}^{ng} \gamma_i(\mathbf{M}) \to \min \tag{10}$$

where: C_1, C_2 - empirical coefficients,

t_e - time of simulation,

$\varphi^{(1)}$ - the tractor roll angle,

v_0 -the initial velocity,

$v_e^{(1)} = \sqrt{\left(\dot{x}^{(1)}(t_e)\right)^2 + \left(\dot{y}^{(1)}(t_e)\right)^2}$ - the total final velocity of the tractor.

The main aim of the optimization problem is to minimize a change in tractor roll angle and the total velocity loss of the articulated vehicle.

4. Modification of Variable Metric Method

The variable metric method which is sometimes called quasi-Newton method is well known and has been used commonly for unconstrained optimization. This method has good theoretical and practical convergence properties [20, 21]. Although these methods were originally developed for small and moderate size dense problems, their modifications based either on sparse, partitioned or limited-memory updates are very efficient for the large-scale sparse problems. However, as in each gradient method, it has some efficiency constraints because in each iteration of the optimization procedure elements of the gradient vector are determined at least once. In majority of works the gradient vector is approximated by difference quotients with use of the finite difference method (e.g. central) with a uniform distribution of the nodes [21]:

$$\Omega_i^{'}(\mathbf{M},\mathbf{q},\dot{\mathbf{q}}) = \frac{\Omega(\boldsymbol{\varphi}(\mathbf{M},\mathbf{q},\dot{\mathbf{q}}) + \mathbf{h}_i) - \Omega(\boldsymbol{\varphi}(\mathbf{M},\mathbf{q},\dot{\mathbf{q}}) - \mathbf{h}_i)}{2h} \tag{11}$$

where:

$\Omega_i^{'}(\mathbf{M},\mathbf{q},\dot{\mathbf{q}})$ - the functional derivative Ω for the condition $\mathbf{M},\mathbf{q},\dot{\mathbf{q}}$ for a *i*-th decisive variable,

$\boldsymbol{\varphi}(\mathbf{M},\mathbf{q},\dot{\mathbf{q}})$ - a function determining the system condition for the determined parameters $\mathbf{M},\mathbf{q},\dot{\mathbf{q}}$,

$$\mathbf{h}_i = \left(\begin{cases} 0 & \text{for} \quad i \neq j \\ h & \text{for} \quad i = j \end{cases} \right)_{j=1,\ldots,m}$$

h - a distance between the adjacent nodes of the discretization grid.

In such anapproach calculating a gradient vector value is connected with multiple calling of the objective function separately for each decisive variable. Thus, determining an objective function value requires that the equations of motion are integrated in the entire time interval of a simulation. If a multi-body system of many degrees of freedom is subject to considerations, optimization computing takes a long time. A final result in the gradient and many non-gradient optimization methods depends strongly on a starting point. A few simulations taking a random starting point and selecting the best solution obtained are made most frequently. Long time of optimization calculations reduces those possibilities and prevents making variant analyses which are important while selecting parameters of system sub-components. Multi-core calculations which are an industrial standard nowadays, can be a direction of the optimization method development what enables to shorten computing time. In the paper a modification of the VMM method was proposed by allotting the calculations to many cores of the processor on which computing is made parallel. The time-consuming way of calculating the gradient vector discussed earlier can be modified by dividing it into a few tasks in which particular elements are determined parallel. The presented way of the division is illustrated in fig. 3 where c_j means j-th processor core, and n_c is a number of the processor cores used in the calculations, f_i means a function computing i-th element of the gradient vector.

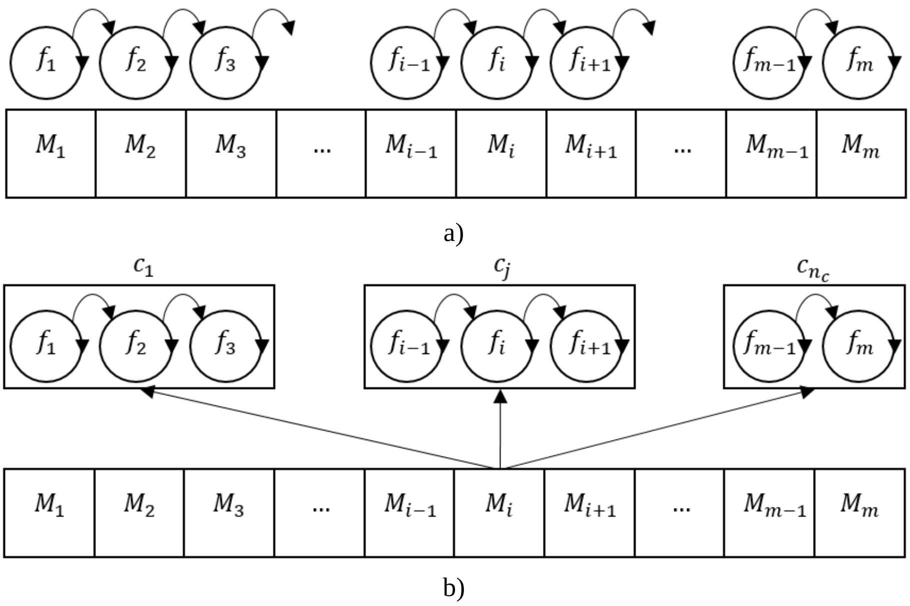

Fig. 3. A sequential way of computing the gradient vector (a), a division of the gradient vector into the elements calculated parallel on many processor cores (b)

Function f_i for specified parameters $\mathbf{M}, \mathbf{q}, \dot{\mathbf{q}}$, determines the appropriate value $\mathbf{h}_i$ and calculates a value of the objective function in point $\boldsymbol{\varphi}(\mathbf{M}, \mathbf{q}, \dot{\mathbf{q}}) + \mathbf{h}_i$. In this paper it has been assumed that that a number of processor cores is the maximum number of parallel processed tasks, and a number of elements performed sequentially on the single processor core c_j depends on a number of a vector of the decisive variables. From the literature studies it results that an attempt to process parallel a greater number of tasks simultaneously than a number of cores can result in general loss of efficiency of a computing unit leading to worsening of the final results, and in consequence, to lengthening of optimization computing time.

5. Verification of numerical model

In order to show adequacy of the mathematical model presented, the computer simulations results were compared to the results obtained in the road tests. An object of the tests was a set with a Mercedes Actros 1840 LS Megaspace F015 tractor to which a Kogel SN 24 P 100/1.060 semitrailer was attached. The multi-body system was equipped with necessary instrumentation: speed sensors of the Correvit firm, placed on the tractor and the semi-trailer, gyroscopes which enable to record yaw velocities and a device to measure a steering wheel revolution angle (fig. 4).

a) b)

Fig. 4 The multi-body system prepared for tests
a) a general view, b) location of the Correvit sensor

A maneuver which corresponds to jerking a steering wheel and may be interpreted as entering a road curve, was analyzed (fig 5). This maneuver is used in testing of vehicle stability and steer-ability (ISO 7401:2003, 2003).

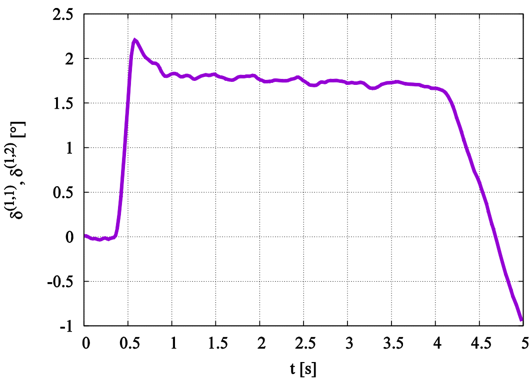

Fig. 5 Course of the steering angle of the front wheel for the maneuver of jerking the steering wheel

The obtained courses of yaw velocities of the tractor and the semitrailer for the road tests and the simulation calculations are presented in fig. 6.

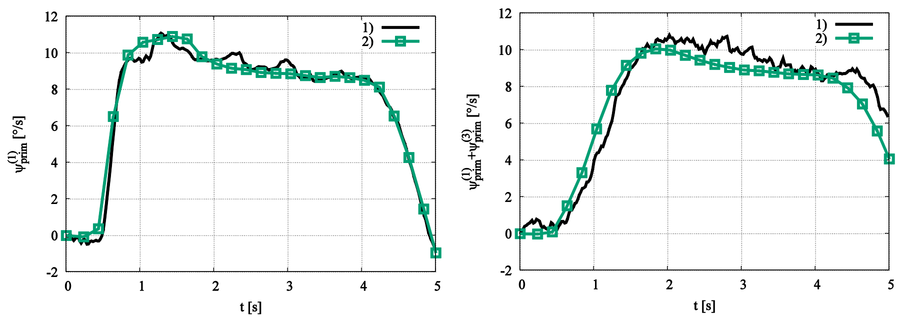

Fig. 6 Comparison of the yaw velocity of a) the tractor, b) the semitrailer obtained from: 1) the road tests, 2) the presented model

In order to assess correctness of the presented model an integral error has been calculated according to the formulae:

$$\varepsilon^{(i)} = \left| \frac{\int_0^{t_e} \left|\dot{\psi}_M^{(i)}\right| dt - \int_0^{t_e} \left|\dot{\psi}_C^{(i)}\right| dt}{\int_0^{t_e} \left|\dot{\psi}_M^{(i)}\right| dt} \right| \cdot 100\% \tag{12}$$

where: $\dot{\psi}_M^{(i)}$ - the yaw velocity obtained from the measurements,

$\dot{\psi}_C^{(i)}$ -the yaw velocity obtained from the model.

The resulting errors calculated according to formulae (4) are shown in table 1.

T a b l e 1

Integral errors obtained from presented model

i	Yaw velocity $\dot{\psi}^{(i)}$	Integral error $\varepsilon^{(i)}$ [%]
1	Tractor	1.2
2	Semitrailer	4.2

It can be seen that percentage error is lower than 5 % for both subsystems. The results presented in table 1 confirm that the presented model of the articulated vehicle owing to good qualitative and quantitative conformity can be used successfully in the optimization computing. Additionally, as shown in [8] the results obtained in the optimization process for a model of a relatively few generalized coordinates can be used in the models of a greater complexity, and even in the models including flexibility of elements e.g. flexibility of a semi-trailer of an articulated vehicle.

5. Numerical simulations

In this chapter optimal braking torques have been calculated using a classical and multi-thread approach of the Variable Metric Method [20, 21]. A lane change maneuver has been taken into account while cornering the articulated vehicle. During this maneuver a rollover of the vehicle occurs. When only the articulated vehicle cornering maneuver is considered the vehicle is stable. The lane change maneuver began at $t_s = 3\,[\mathrm{s}]$ of the simulation and at the same time additional braking torques started. These torques are applied until the end of the simulation ($t_e = 6$ [s]). Vehicle initial velocity $v_0 = 45$ [km/h] is assumed. Physical parameters of the articulated vehicle are taken from [6]. Interpolation of the braking torque has been performed for n=7 discrete values. In the optimization process it has been assumed that the braking torques will be determined individually for each tractor wheel of the articulated vehicle, whereas the same values of the braking torques at any time will act on the groups of the semi-trailer wheels. (fig. 7).

Fig. 7. Numbering of the articulated vehicle wheels

Bulrish-Stoer-Deuflhard [21] method with an adaptive step size has been used to integrate equations of motion. Implementation of the mathematical model and optimization algorithms was made in the proprietary program in language C++, which as opposed to other languages of a high level, such as e.g. C#, is independent of hardware architecture and an operating system. Fig. 8 shows a course of a steering angle of the articulated vehicle front wheels applied during optimization.

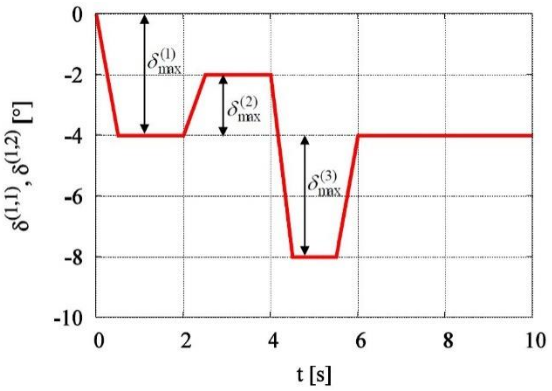

Fig. 8. A course of the steering angle of the articulated vehicle

The optimization computing for the described problem of a selection of the braking torques acting on the articulated vehicle wheels in a critical situation was tested on four equipment platforms with a different processor, an operating system and an implementation language complier. Two computers with the Microsoft Windows operating system, and two ones with the OSX system and two four- and eight-core machines were used respectively in the tests. The parameters of the computing units are presented in table 2.

Table 2

Parameters of the computing units used in the tests

Name	*OS*	*Parameters*	*C++ compiler*
OS X i7	OS X El Capitan 10.11.3	Intel Core i7 CPU 2,80 GHz, 16 GB RAM	Apple LLVM version 7.0.2 (clang-700.1.81)
OS X i5	OS X El Capitan 10.11.3	Intel Core i5 CPU 2,60 GHz, 8 GB RAM	Apple LLVM version 7.0.2 (clang-700.1.81)
Win i7	Microsoft Windows 10	Intel Core i7-3632QM CPU 2,20 GHz, 8 GB RAM	Microsoft C/C++ Version 19.00.23506
Win i5	Microsoft Windows 10	Intel Core i5-3230M CPU 2,60 GHz, 12,0 GB RAM	Microsoft C/C++ Version 19.00.23506

Examples of the results obtained from optimization are presented in fig.8. It shows courses of the tractor roll angle, the vertical displacement and its trajectory. The figures also show the results of the simulations for the lane following (cornering) maneuver (LF) and those with the lane change maneuver before optimization (BO).Figure 4d shows the courses of the optimal braking torques for the tractor and the trailer wheels 1-6.

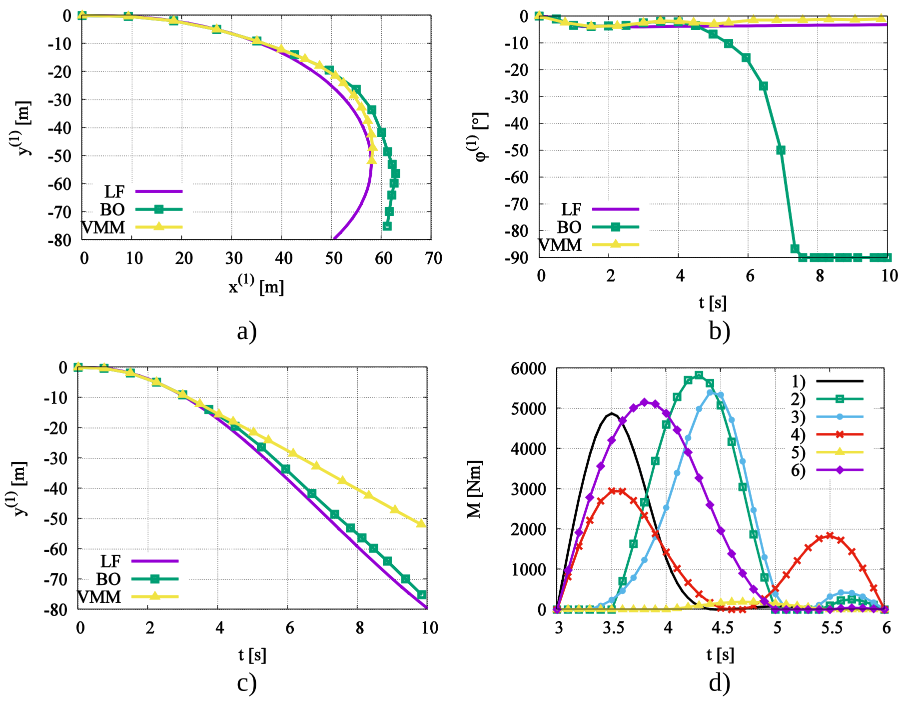

Fig. 8. Courses of the articulated vehicle a) the tractor trajectory, b)the roll angle, c) the vertical displacement, d) optimal braking torques

It can be seen that the dynamic optimization method in question provides a solution which allows the vehicle to maintain stability during the maneuver. In the subsequent diagrams (fig. 9 and 10) velocities of the vehicle while performing the maneuver are presented.

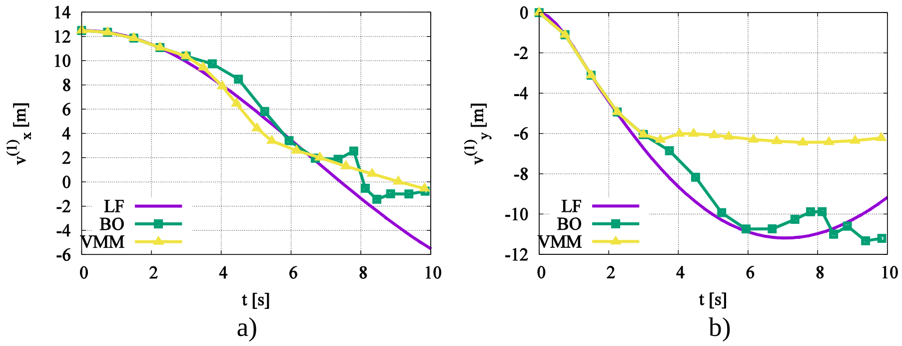

Fig. 9. The courses of the articulated vehicle velocity a) longitudinal, b) lateral

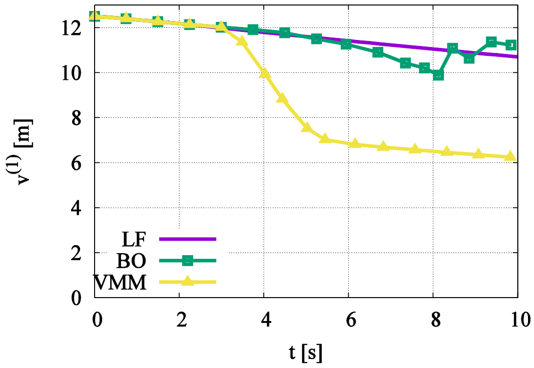

Fig. 10. The courses of the total velocity of the articulated vehicle tractor

It can be noticed that in the case in question braking moments resulting in reducing the vehicle velocity resultant of almost a half of the initial value should be generated in order to perform the maneuver safely.

In spite of obtaining the results in a form the braking moments the vehicle wheels providing safe performance of the maneuver in question, the optimization process is relatively long-lasting. Measurements of time calculations with use of parallel computing are presented in the further part of the paper. For this purpose there were 12 tests conducted for each computing variant understood as conducting an entire optimization process on each computing unit with use from 1 to n_c of the processor cores (depending on the equipment architecture). Then, the extreme results were rejected, and 10 remaining simulations were used to calculate the average values. During the tests the following parameters were recorded:

- average time of optimization process calculations,
- a number of objective function calls,
- average time of an execution of a single objective function.

The results obtained during computing are presented in tables from 3 to 6.

Table 3

Averaged results of optimization with use of many processor cores in system OS X i5

Number of cores	*Optimization time*	*Number of objective function calls*	*Execution average time*
1	494.407	930.400	0.544
2	368.303	1100.670	0.338
3	231.543	712.667	0.328
4	213.654	715.800	0.302

T a b l e 4

Averaged results of optimization with use of many processor cores in system OS X i7

Number of cores	*Optimization time*	*Number of objective function calls*	*Execution average time*
1	324.210	709.500	0.440
2	264.861	1076.333	0.247
3	183.730	991.000	0.186
4	242.027	1345.333	0.180
5	164.996	1076.667	0.155
6	174.096	1155.667	0.148
7	150.893	906.000	0.168
8	233.575	1115.100	0.209

T a b l e 5

Averaged results of optimization with use of many processor cores in system Win i5

Number of cores	*Optimization time*	*Number of objective function calls*	*Execution average time*
1	1142.200	1175.000	0.964
2	740.793	1278.667	0.589
3	383.222	750.600	0.508
4	413.695	951.900	0.441

T a b l e 6

Averaged results of optimization with use of many processor cores in system Win i7

Number of cores	*Optimization time*	*Number of objective function calls*	*Execution average time*
1	983.969	1006.900	0.987
2	628.124	1141.667	0.543
3	334.426	958.000	0.378
4	354.589	949.667	0.374
5	348.595	995.667	0.346
6	296.256	827.667	0.342
7	597.218	1656.667	0.359
8	398.173	1016.600	0.398

The averaged time of the single objective function execution deems to be the most reliable indicator of the real execution time, because the tests were conducted starting from random starting points, and a number of the objective function calls was different depending on the starting point and on a value of the final extreme obtained. The average time of execution for one call of the objective function is presented in fig.11.

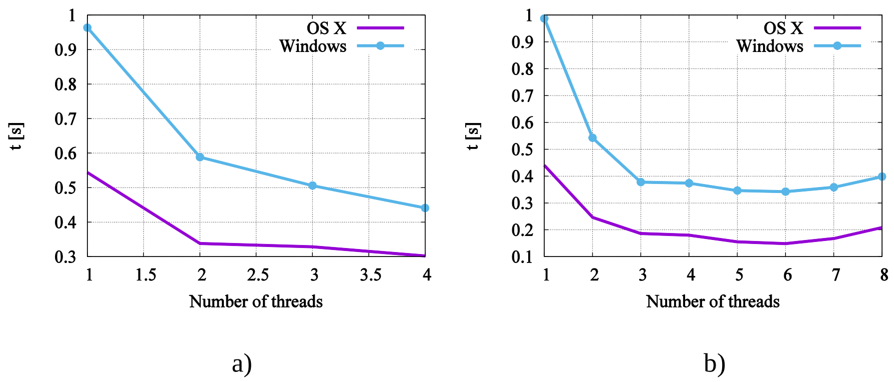

Fig. 11. The average time of execution for one call of the objective function for architecture a) i5 i b) i7

A significant improvement in effectiveness of the optimization method can be noticed with implementing parallel computing. Significant shortening of the computing time occurs even at use of only one additional processor core. Independently of an operating system splitting a vector of the decisive variables into six parallel tasks for architecture i7 is optimal in the case in question. An increase in a number of the cores used above this value resulted in lengthening of computing time of a single objective value. According to the authors it may be caused by too low number of atom tasks for one processor core in relation to time needed to synchronize the threads. For optimization problems requiring a great number of elements in the decisive variable vector it deems to be, certainly, justified to use all available cores, and even performing calculations on the general-purpose computing on graphics processing units (GPGPU).

In order to compare quantitatively the results obtained in the classical optimization process with the algorithm providing parallel processing, a coefficient which specifies an improvement of the computing time determined according to the formula was implemented:

$$\Delta_i = \frac{R_1}{R_i}, i = 2, \ldots, n_c \quad (13)$$

where:

R_1 - average execution time for one objective function call with use of one core (a sequential algorithm),

R_i - average execution time for one objective function call with use of i cores.

The obtained results showing an increase in efficiency with use of parallel computing of values of gradient vector elements are presented in fig. 12a for architecture i5 and 12b for architecture i7.

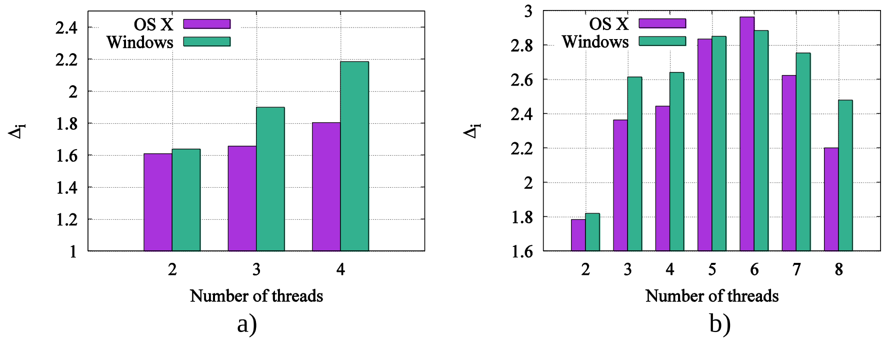

Fig. 12. An increase in efficiency by implementing parallel processing for architecture: a) i5, b) i7

The presented results indicate that independently of the architecture of the computing unit there is an increase in efficiency when parallel computing ($\Delta_i > 1$) is used. Additionally, in almost all cases considered, a greater increase in efficiency was recorded for the machines with the Microsoft Windows 10 operating system. However, it should be emphasized that single core optimization computing made in the OS X system was more efficient than in the Windows system. The presented results confirm the previous conclusion that in the case in question use of six cores is optimal. For this case almost triple shortening of optimization computing time was recorded comparing to sequential calculations on one processor core.

5. Conclusions

Majority of articulated vehicle rollovers occur whilst cornering. Due to the higher centre of gravity, and low rollover threshold, entering a corner at excessive speeds encourages the vehicle to lean and thus rollover. It is very important to determine braking torques which restore stability of the vehicle. In the presented paper the problem of controlling brakes during cornering has been formulated and solved as dynamic optimisation problem. The variable metric method has been used in order to obtain the optimal braking torques. After applying the results obtained from the dynamic optimisation, vehicle stability is restored and the roll angle is significantly smaller than before optimisation. The dynamic optimization can be made on basis of simplified models of vehicles. An advantage of such a solution is shortening of computing time, and the results obtained for the simplified models can be used for models of a higher degree of complexity [8]. However, even in this case computing time is relatively long. In this paper an attempt was made to reduce additionally optimization computing time by implementing a modification in a way of determining the values of the elements of the gradient vector, which is basis of the variable metric method (and other gradient methods). This modification consisted in use of multi-core computing and it enabled to determine the objective function value in a parallel way; an operation which is the most time-consuming part of the optimization procedure. It allowed to shorten computing time, and depending on a number of the processor cores used the gain was significant. In the case in question almost triple shortening of the optimization computing time was achieved comparing to computing by the classical variant of the VMM method. The presented algorithm of the proceedings can be used successfully in other gradient optimization methods, when it is used for problems in which time of determining single value of the objective function is dominant. .

References

[1] Yedavalli R.K., Robust stability and control of multi-body ground vehicles with uncertain dynamics and failures, Technical Report, Ohio State University Research Foundation, Ohio, 2010

[2] Yao Z. et al: Dynamic simulation for the rollover stability performances of articulated vehi-cles, Journal of Automobile Engineering pp. 771-783 (2014)

[3] Forkenbrock G. J., Garrot W., Heitz M., O'Harra B. C., A com-prehensive experimental examination of test maneuvers that may induce on-road, untripped, light vehicle rollover - phase iv of nhtsa's light vehicle rollover research program, Technical Report. National Highway Traffic Safety Administration, 2002

[4] Huang, H.H., Controller design for stability and rollover prevention of multi-body ground vehicles with uncertain dynamics and faults, PhD Thesis, Graduate School of The Ohio State University, Ohio, 2009

[5] Warwas K., Augustynek K., Dynamic optimisation of articulated vehicle motion for control of stability in critical situation," IDAACS'2015: 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Vol. 1, 2015, pp. 232-237

[6] D. Oberoi, Enhancing roll stability and directional performance of articulated heavy vehicles based on anti-roll control and design optimization, PhD Thesis, University of Ontario Institute of Technology, 2011

[7] S.-B. Lu, Y.-N. Li, S.-B. Choi, Contribution of chassis key subsystems to rollover stability control, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering April, vol. 226 no. 4, 2012, pp. 479-493

[8] Warwas K., Analiza i sterowanie ruchem pojazdów wieloczłonowych z uwzględnieniem podatności elementów, Praca doktorska, Akademia Techniczno-Humanistyczna, Bielsko-Biała, 2008.

[9] Reinders J.,Jeffers J., High Performance Parallelism Pearls Volume One: Multicore and Many-core, Elsevier, 2015

[10] J. Leng,W. Sharrock, Handbook of Research on Computational Science and Engineering: Theory and Practice, IGI Global, 2012

[11] Negrut D., Tasora A., Mazhar H., Heyn T., Hahn P., Leveraging parallel computing in multibody dynamics, Multibody System Dynamics, Volume 27, Issue 1, 2012, pp. 95-117

[12] Eberhard, P., Dignath, F., Kübler, L.: Parallel evolutionary optimization of multibody systems with application to railway dynamics, Multibody System Dynamics, 9 (2), 2003, pp.143–164

[13] Umbarkar A.J., Joshi M.S., Hong Wei-Chiang, Multithreaded Parallel Dual Population Genetic Algorithm (MPDPGA) for unconstrained function

optimizations on multi-core system, Applied Mathematics and Computation, Volume 243, 2014, pp. 936-949

[14] Augustynek K., Warwas K., Polański A., Application of the genetic algorithms and distributed computing in task of the reduction of vibrations of a satellite, CMS'09: 7 th Conference Computer Methods and Systems, 2009, pp. 237-242

[15] A. Jutty K, M. S. Bhat, D. Ghose, Performance of Parallel Shooting Method for Closed Loop Guidance of an Optimal Launch Vehicle Trajectory, Optimization and Engineering, Volume 1, Issue 4, 2000, pp 399-435.

[16] Kubica B.J., Woźniak A., Optimization of the Multi-Threaded Interval Algorithm for the Pareto SetComputation, Journal of Telecommunications and Information Technology, Vol. 1, 2010, pp. 70-75

[17] Bauchau O.A., Flexible Multibody Dynamics, Solid Mechanics and Its Applications, Springer Netherlands, 2011

[18] Chong E., Zak S., An Introduction to Optimization, 4th Edition, Wiley, 2013

[19] Rao S., Engineering Optimization: Theory and Practice, 4th Edition, Wiley, 2009

[20] Luksan L., Spedicato E., Variable metric methods for unconstrained optimization and nonlinear least squares, Journal of Computational and Applied Mathematics, Elsevier, 2000, pp. 61-95

[21] Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes 3rd Edition: The Art of Scientific Computing, Cambridge University Press, Cambridge (2007)