7. Гаскаров Д. В., Голинкевич Т. А., Мозгалевский А. В. Прогнозирование технического состояния и надежности радиоэлектронной аппаратуры / Под ред. Т.А.Голинкезича. – М.: Сов. радио, 1974.

Ю. Б. Прибылев, Д. П. Пашков, Л. М. Сакович

ВЛИЯНИЕ ВРЕМЕННЫХ ХАРАКТЕРИСТИК СИСТЕМЫ ВООРУЖЕНИЙ НА ЗАГРЯЗНЕНИЕ СРЕДЫ

Рассмотрено влияние технологического уровня вооружений и военной техники на экологические последствия их применения. Получено условие соблюдения временного баланса ракетного комплекса.

Ключевые слова: *сложная техническая система, ракетный комплекс, временной баланс.*

Y. Pribylev, D. Pashkov, L. Sakovich

EFFECT OF TIME CHARACTERISTICS OF WEAPONS ON POLLUTION PROTECTION

The influence of the process of arms and military equipment to the environmental consequences of their use are considered. Time balance of anti-aircraft missile is described.

Keywords: *complex technical system, missile complex, maintenance performance.*

UDC 629.331

Warwas Kornel

PARALLEL GENETIC ALGORITHM WITH ACTOR MODEL APPROACH TO RESTORE STABILITY OF AN ARTICULATED VEHICLE

The paper presents an application of an actor model to control braking torques on wheels of an articulated vehicle in an untripped rollover manoeuvre. The numerical model of the articulated vehicle and dynamic optimisation have been used to calculate appropriate braking torques for each wheel in order to restore stability. The optimisation problem requires the equations of motion to be integrated at each optimisation step and it is a time-consuming task. Therefore, parallel computing with using actor model system has been proposed. The actor model system has been implemented in genetic algorithm. In the paper, formulation of genetic algorithm with the actor system and results obtained from dynamic optimisation have been presented and compared.

Key words*: parallel computing, actor model, genetic algorithm, optimisation, articulated vehicle.*

Introduction. The dynamic optimisation of sophisticated physical systems such as multibody system and articulated vehicle is an excessively time-consuming task. Improvement of the optimisation calculation time is a subject of many papers [1, 2, 3]. Parallel and distributed systems are often used in order to improve the efficiency of calculations [1, 2, 4, 5]. Some of the algorithms allow splitting computational effort in separate threads, processes or cluster's nodes in a natural way. The approach used in order to formulate model of the system and implementation of the optimization process are not without significance. Currently, the development of computer hardware and software allows to release from the monolithic architecture to micro-services approach in which part of the business logic can be separately processed and those parts can communicate with each other [6]. According to Reactive Manifesto [7, 8], the ability to implement software message driven in responsive, resilient and elastic way is very important and it is the subject of many papers [9, 10, 11]. Also these modifications allow significantly reducing the time of calculation by parallel processing and providing high scalability of the system. One of ways such as implementation of the system and calculations is called the actor model approach [4, 12, 13]. The actor model for concurrent and parallel programming is gaining popularity due to its high level of abstraction and its ability to make efficient use of multicore and multiprocessor machines. The actor model describes calculation process as a result of interaction between active objects (actors). Actors interact with each other by asynchronous messages passing. Each actor can process a received message according to implemented behaviour, create other actors, forward the message to other actor or wait for the new message. There are many existing Actor Model implementations such as Akka, CAF, Theron, Pykka [8, 14, 15]. Some of actor model systems can be deployed in remote environment such as computational grid or cloud systems.

In this paper genetic algorithm with the actor model approach has been presented. Process of optimisation concerning manoeuvring of the articulated vehicle has been considered. The aim of the optimisation calculations is to maintain the stability of vehicle and to prevent before its rollover during manoeuvring. Rollover accidents of articulated vehicles are especially violent and cause greater damage and injury than other accidents. The relatively low roll stability of trucks promotes rollovers and contributes to the large number of vehicle accidents [16]. 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 which can provide a driver with greater control [3, 17, 18, 19]. Lots of automotive companies introduce their own solutions which improve vehicle stability. The appropriate 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 to reduce the risk of vehicle instability while driving on a slippery curve or taking an evasive action, to diminish the potential for jackknifing and drift-out conditions through a selection of braking systems of the tractor and an application of the trailer brakes which provide both Roll Stability Control (RSC) functionality and 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 have been mentioned such as active front steering (AFS), active braking (AB), and active suspension (AS) control for rollover prevention. They have been used based on a full vehicle model [20]. The design of virtual models allows the number of experimental road tests of real vehicles to be decreased [3, 19]. Presented in this paper method is based on the virtual model of the articulated vehicle and allows controlling 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 optimisation task.

2Mathematical model of an articulated vehicle. The model of the articulated vehicle was formulated as a system of rigid bodies: a tractor, a fifth wheel, a semi-trailer, forming an open kinematic chain (fig. 1).

It is assumed that the tractor is a rigid body, whose motion is described by means of six generalized coordinates, the fifth wheel has one degree of freedom (a pitch angle) in relation to the tractor, the semi-trailer has one degree of freedom (an inclination angle) with respect to the fifth wheel. Wheels are connected with the tractor and the semi-trailer and each has one degree of freedom. The tractor is set on four wheels and the semi-trailer has six wheels. Suspension stiffness has been reduced to the contact point of the tire with the road. Additionally, the model contains generalized coordinates which are the front wheels’ steering angles of the tractor. 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 $\tilde{\mathbf{q}}_T^{(1)} = \begin{bmatrix} x^{(1)} & y^{(1)} & z^{(1)} & \psi^{(1)} & \theta^{(1)} & \varphi^{(1)} \end{bmatrix}^T$ – generalized coordinates vector of trailer,

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

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

$\tilde{\mathbf{q}}_F^{(2)} = \begin{bmatrix} \theta^{(2)} \end{bmatrix}^T$ – generalized coordinates vector of fifth wheel,

$\tilde{\mathbf{q}}_S^{(3)} = \begin{bmatrix} \psi^{(3)} \end{bmatrix}^T$ – generalized coordinates vector of semi-trailer,

$\tilde{\mathbf{q}}_{SW}^{(3)} = \begin{bmatrix} \theta^{(3,1)} & \theta^{(3,2)} & \theta^{(3,3)} & \theta^{(3,4)} & \theta^{(3,5)} & \theta^{(3,6)} \end{bmatrix}^T$ – generalized coordinates vector of semi-trailer wheels,

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

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

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

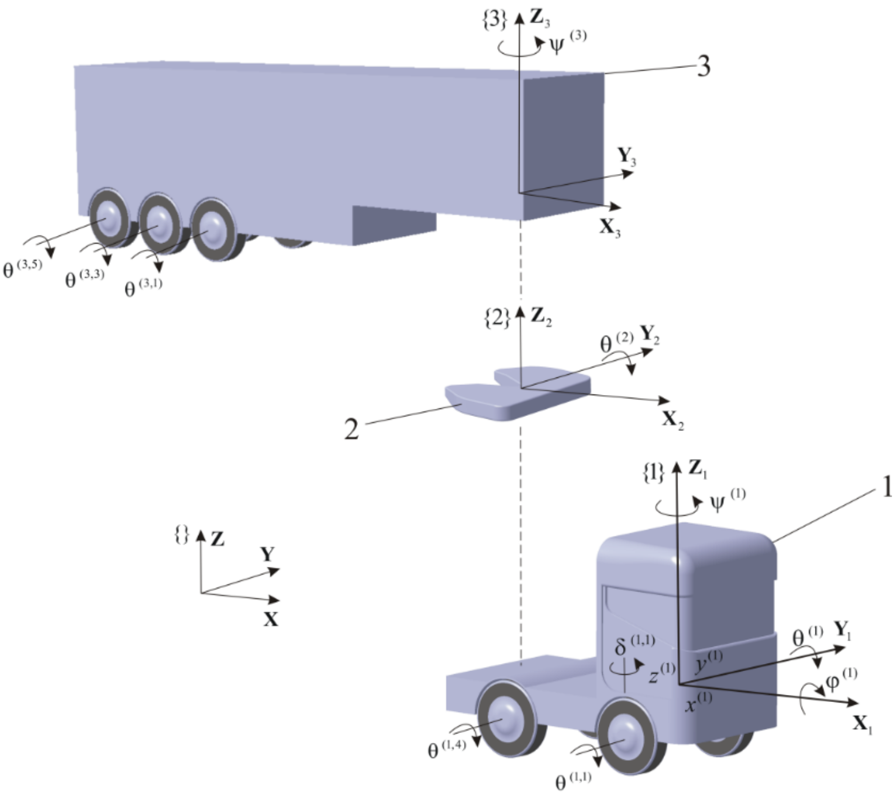


Figure 1. The model of the articulated vehicle: 1) a tractor, 2) a fifth wheel, 3) a semi-trailer

After transformations dynamic equations of motion of the *i*-th subsystem of the articulated vehicle can be written in the general form

$$\mathbf{A}^{(i)}\ddot{\mathbf{q}}^{(i)} = \mathbf{f}^{(i)}, \tag{2}$$

where $\mathbf{A}^{(i)}$ – mass matrix,

$\mathbf{f}^{(i)}$ – vector of external, Coriolis, centrifugal and gravity forces.

Equations of motion of the particular subsystems of the articulated vehicle can be described as follow:

- subsystem 1 – tractor

$$\mathbf{A}^{(1)} = \begin{bmatrix} \mathbf{A}^{(1)}_{T,T} & \mathbf{A}^{(1)}_{T,TS} & \mathbf{A}^{(1)}_{T,TW} \\ \mathbf{A}^{(1)}_{TS,T} & \mathbf{A}^{(1)}_{TS,TS} & \mathbf{A}^{(1)}_{TS,TW} \\ \mathbf{A}^{(1)}_{TW,T} & \mathbf{A}^{(1)}_{TW,TS} & \mathbf{A}^{(1)}_{TW,TW} \end{bmatrix}, \ \mathbf{q}^{(1)} = \begin{bmatrix} \tilde{\mathbf{q}}^{(1)}_{T} \\ \tilde{\mathbf{q}}^{(1)}_{TS} \\ \tilde{\mathbf{q}}^{(1)}_{TW} \end{bmatrix}, \ \mathbf{f}^{(1)} = \begin{bmatrix} \mathbf{f}^{(1)}_{T} \\ \mathbf{f}^{(1)}_{TS} \\ \mathbf{f}^{(1)}_{TW} \end{bmatrix}$$

- subsystem 2 – fifth wheel

$$\mathbf{A}^{(2)} = \begin{bmatrix} \mathbf{A}^{(2)}_{T,T} & \mathbf{A}^{(2)}_{T,F} \\ \mathbf{A}^{(2)}_{F,T} & \mathbf{A}^{(2)}_{F,F} \end{bmatrix}, \ \mathbf{q}^{(2)} = \begin{bmatrix} \tilde{\mathbf{q}}^{(1)}_{T} \\ \tilde{\mathbf{q}}^{(2)}_{F} \end{bmatrix}, \ \mathbf{f}^{(2)} = \begin{bmatrix} \mathbf{f}^{(2)}_{T} \\ \mathbf{f}^{(2)}_{F} \end{bmatrix},$$

- subsystem 3 – semi-trailer

$$\mathbf{A}^{(3)} = \begin{bmatrix} \mathbf{A}^{(3)}_{T,T} & \mathbf{A}^{(3)}_{T,F} & \mathbf{A}^{(3)}_{T,S} & \mathbf{A}^{(3)}_{T,SW} \\ \mathbf{A}^{(3)}_{F,T} & \mathbf{A}^{(3)}_{F,F} & \mathbf{A}^{(3)}_{F,S} & \mathbf{A}^{(3)}_{F,SW} \\ \mathbf{A}^{(3)}_{S,T} & \mathbf{A}^{(3)}_{S,F} & \mathbf{A}^{(3)}_{S,S} & \mathbf{A}^{(3)}_{S,SW} \\ \mathbf{A}^{(3)}_{SW,T} & \mathbf{A}^{(3)}_{SW,F} & \mathbf{A}^{(3)}_{SW,S} & \mathbf{A}^{(3)}_{SW,SW} \end{bmatrix}, \ \mathbf{q}^{(3)} = \begin{bmatrix} \tilde{\mathbf{q}}^{(1)}_{T} \\ \tilde{\mathbf{q}}^{(2)}_{F} \\ \tilde{\mathbf{q}}^{(3)}_{S} \\ \tilde{\mathbf{q}}^{(3)}_{SW} \end{bmatrix}, \ \mathbf{f}^{(3)} = \begin{bmatrix} \mathbf{f}^{(3)}_{T} \\ \mathbf{f}^{(3)}_{F} \\ \mathbf{f}^{(3)}_{S} \\ \mathbf{f}^{(3)}_{SW} \end{bmatrix}$$

Dynamic equations of motion of the articulated vehicle have to take into account constraint equations which ensues from kinematic input that describes course of the steering angle of the front wheels. Finally, dynamic equations of motion of the articulated vehicle take a form [3, 21, 22]

$$\begin{gathered} \mathbf{A}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}\mathbf{r} = \mathbf{f} \\ \mathbf{\Phi}^{T}_{\mathbf{q}}\ddot{\mathbf{q}} = \mathbf{w} \end{gathered} \qquad (3)$$

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

$\mathbf{\Phi}_{\mathbf{q}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ 1 & 0 \\ 0 & 1 \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ – constraints matrix,

$\mathbf{f} = \begin{bmatrix} \mathbf{f}^{(1)}_{T} + \mathbf{f}^{(2)}_{T} + \mathbf{f}^{(3)}_{T} \\ \mathbf{f}^{(2)}_{F} + \mathbf{f}^{(3)}_{F} \\ \mathbf{f}^{(3)}_{S} \\ \mathbf{f}^{(1)}_{TS} \\ \mathbf{f}^{(1)}_{TW} \\ \mathbf{f}^{(3)}_{SW} \end{bmatrix}$ – vector of external, Coriolis and centrifugal forces,

$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$ – vector of unknown constraint reactions,

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

r_1, r_2 – torques on the suspension connected to the wheels.

The details of the procedure which lead to formation of equation (3) with a description of elements in the matrix $\mathbf{A}$ and the vector $\mathbf{f}$ are presented in [3]. Articulated vehicle model has been verified by comparing the results obtained from the model with those obtained from the road tests. During experiment the J-turn maneouvre described in international standard ISO 7401:2003 has been investigated. Response of the articulated vehicle to sudden step steering angle has been analysed (fig. 2).

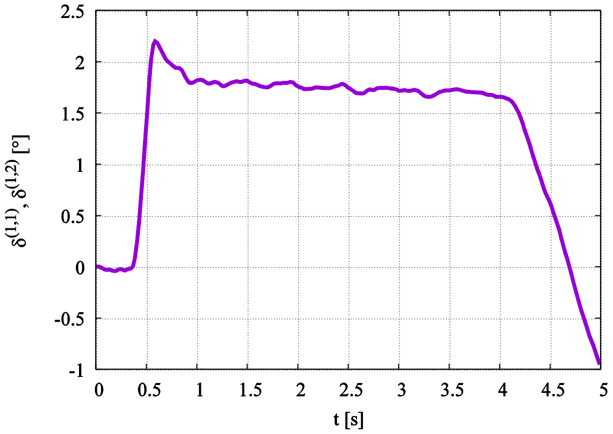


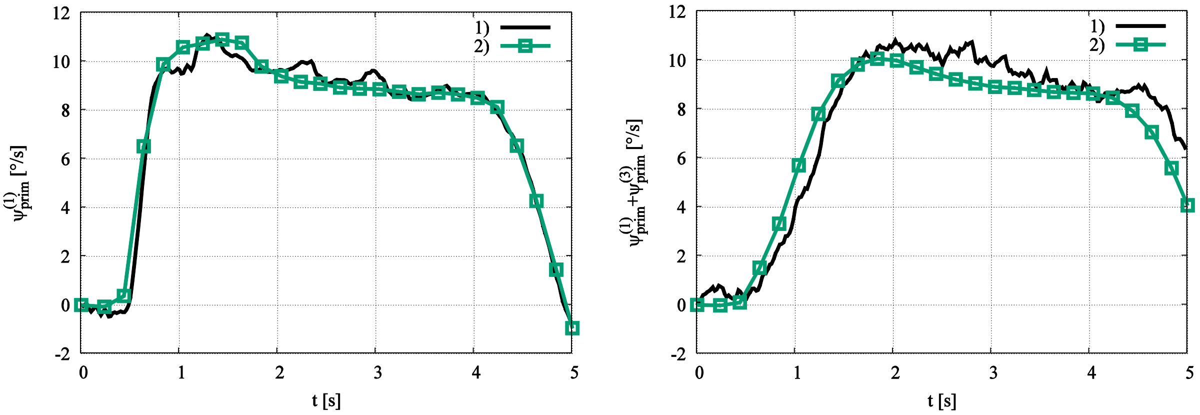
Figure 2. Course of the front wheels steering angle used during road tests and simulations

Tractor unit Mercedes Actros 1840 LS Megaspace F015 together with box semi-trailer Kogel SN 24 P 100 / 1.060 have been tested (fig. 3a). Correvit velocity sensors (fig. 3b) and gyroscopes have been used in order to measure yaw velocity of the vehicle and steering angle of the front wheels.

a) b)

Figure 3. Tested articulated vehicle:
a) general view, b) location of the Correvit sensor

Comparison of the tractor and the semi-trailer yaw velocity courses obtained from the model and experiment has been shown in fig. 4.



a) b)

Figure 4. Comparison of the yaw velocity of a) the tractor, b) the semi-trailer obtained from: 1) road tests, 2) presented model

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

$$\varepsilon^{(i)} = \left| \frac{\int_0^{t_e} \left|\dot{\psi}_M^{(i)}\right| dt - \int_0^{t_e} \left|\dot{\psi}_C^{(i)}\right| dt}{\int_0^{t_e} \left|\dot{\psi}_M^{(i)}\right| dt} \right| \cdot 100\% , \qquad (4)$$

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

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

The resulting errors calculated according to above formulae have been shown in the table 1.

Table 1

Optimisation methods parameters

	Yaw velocity $\dot{\psi}^{(i)}$	Integral error $\varepsilon^{(i)}$ [%]
	Tractor	1.2
	Semi-trailer	4.2

Acceptable correspondence of the results has been achieved. Relative integral error calculated for both courses does not exceed 5 %.

Formulation of the vehicle optimisation problem

Articulated vehicles rollover is one of the most dangerous road manoeuvres. This situation happens mostly during the unforeseen lane-change manoeuvre [16] of the vehicle. Such manoeuvre has been performed when the preplanned vehicle trajectory would collide with an obstacle. When the obstacle is detected, the trajectory is translated to other traffic lane, as shown in fig. 5 in order to avoid collisions with the obstacle.

Stability of the articulated vehicle can be restored by an appropriate control of braking torques applied to each wheel of the vehicle. Let us consider a vector of braking torque discrete values $\mathbf{M}^{(i)}$ of the *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 discrete values of the braking torques of wheels and can be written in the form:

$$\mathbf{M} = \left[\mathbf{M}^{(1)} \quad \ldots \quad \mathbf{M}^{(i)} \quad \ldots \quad \mathbf{M}^{(n_w)}\right]^T = \left(M_j\right)_{j=1,\ldots,m} , \qquad (5)$$

where m – number of decisive variables,

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

n – number of discrete values of the braking torque.

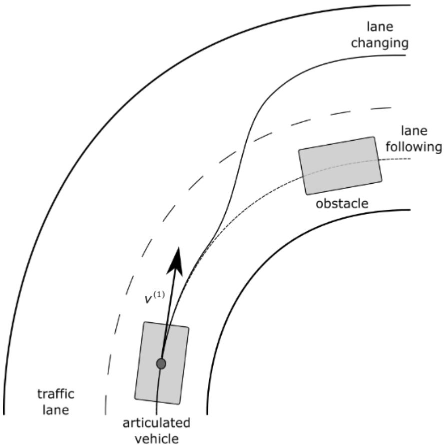


Figure 5. Lane changing and lane following manoeuvre during cornering

The stability conditions can be assured by solution of dynamic optimisation problem which can be presented in the general form [23]

$$\Omega(\mathbf{M}, \mathbf{X}_1, \ldots, \mathbf{X}_{n_r}) \to \min, \tag{6}$$

where Ω – objective function,

$\mathbf{X}_i$ – objective function dependency

n_r – number of objective function dependencies

Calculations of the objective function (6) require integration of model dynamic equations (3). In the presented problem braking torques calculated for a fixed initial vehicle velocity and front wheels steering angle have to fulfil following conditions: the articulated vehicle cannot lose stability during the manoeuvre, total velocity loss has to be as small as possible,

Above assumptions are taken into account in the objective function and also in optimisation constraints. The stability conditions can be assured by minimizing the functional

$$\tilde{\Omega}(\mathbf{M}, \mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{t_e}\left(C_1 \int_0^{t_e} \left(\varphi^{(1)}\right)^2 dt + C_2\left(v_0 - v_e\right)\right) \to \min, \tag{7}$$

where C_1, C_2 – empirical coefficients,

t_e – time of simulation,

v_0 – initial velocity,

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

In the considered optimisation problem, inequality constraints can be written as follows

$$M_{\min,j} \le M_j \le M_{\max,j}, \tag{8}$$

where $M_{\min,j}, M_{\max,j}$ describe minimum and maximum value of the braking torque M_j. These conditions can be included through additional inequality constrains [23, 24], which can be written in the following form:

$$g_i(\mathbf{M}) \le 0, \tag{9}$$

where $i = 1, \ldots, n_g$,

n_g – number of the inequality constraints.

The constraints can be included in the objective function by the external penalty function [24]:

$$\gamma_i(\mathbf{M}) = \begin{cases} 0 & \text{for} \quad g_i(\mathbf{M}) \le 0 \\ C_{1,i} \exp\left(C_{2,i} g_i(\mathbf{M})\right) & \text{for} \quad g_i(\mathbf{M}) > 0 \end{cases} \tag{10}$$

where $C_{1,i}$, $C_{2,i}$ are empirical weights. Finally, objective function has a form:

$$\Omega(\mathbf{M}, \mathbf{q}, \dot{\mathbf{q}}) = \tilde{\Omega}(\mathbf{M}, \mathbf{q}, \dot{\mathbf{q}}) + \sum_{i=1}^{n_g} \gamma_i(\mathbf{M}) \to \min . \tag{11}$$

The classical genetic algorithm and genetic algorithm with the actor model approach method have been applied in order to solve formulated minimization problem [23, 25]. In the next paragraph genetic algorithm with the actor model approach method will be presented in details.

Genetic algorithm with actor model. Actors are objects which encapsulate state, behaviour and they can communicate exclusively by exchanging messages which are placed into the recipient's mailbox (fig. 6).

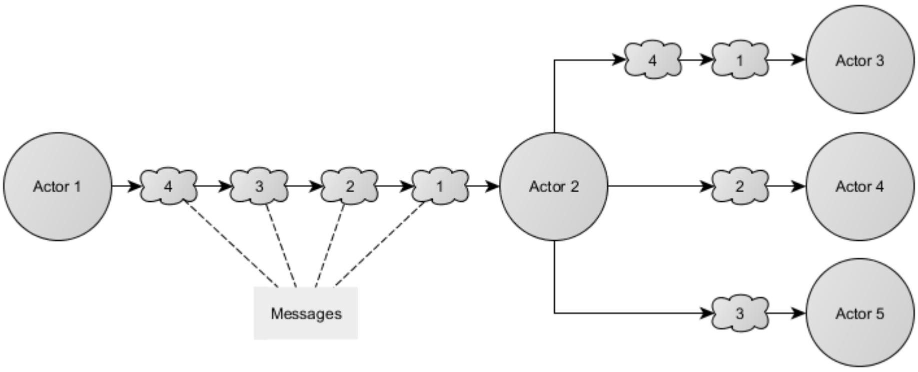


Figure 6. Messages exchanging in actor model system

One actor can split up its tasks and delegate until they become small enough to be handled in one piece in order to build fault-tolerant and hierarchical system. Each actor can be treated as separate thread on one computer or a process on machine in cluster of distributed systems. Such approach is therefore suitable for designing and development of parallel and distributed information systems. Actor model basis on event-driven system in which actors process events and generate responses or more requests in asynchronous way and message passing is network transparent. In this paper CAF framework [14, 15] has been applied in order to implement genetic algorithm with actor model approach. CAF allows to transparently connect actors running on different machines and operating systems via the network. It integrates multiple computing devices such as multicore CPUs, GPGPUs, and even embedded hardware. In presented approach first actor (system) creates next actor (generation) which is responsible for managing genetic algorithm generations (fig. 7).

Generation actor produces many coworkers (crossover operators) which perform next operation in non-blocking and asynchronous way. Number of those child actors depends on number of individuals in population. When crossover operation is finished each actor produces new actor which is responsible for mutation operation. Individuals obtained from mutation are forwarded to actors which evaluate objective function value. This operation is time consuming because it is necessary to integrate dynamic equation of motion of the system. The main advantage of such approach is that the calculation of the objective function can be performed in parallel and non-blocking manner. Thereafter, actor waits until calculation of the objective function will be finished by all actors. It collects results and creates new parent's population according to natural selection rule. These steps are repeated until stop condition is not satisfied. Messages which are sent during one step of presented approach are shown in fig. 8.

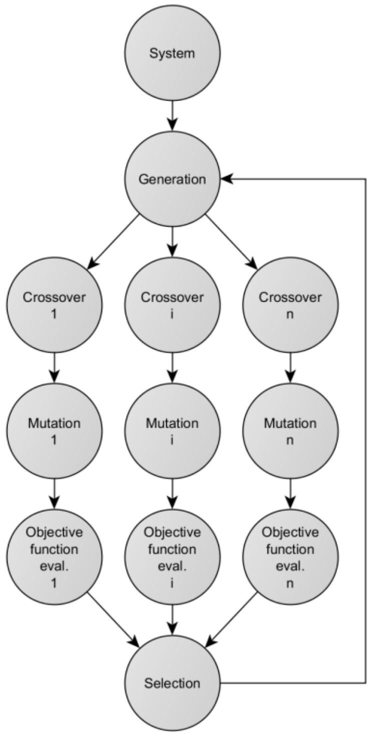


Figure 7. Actors in genetic algorithm

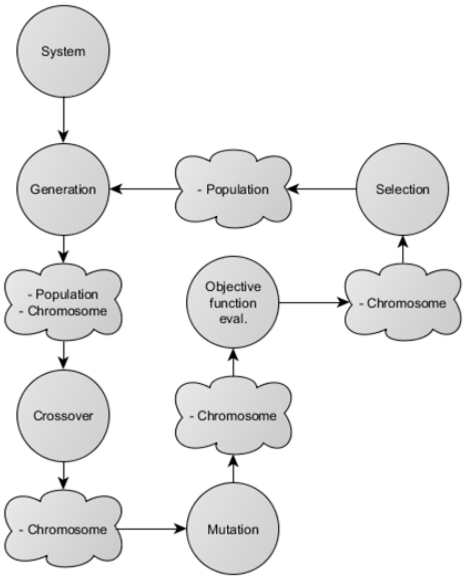


Figure 8. Actors in genetic algorithm

It is assumed real-number representation of genes in chromosomes and the following genetic operators have been used [25]: hybrid selection consists of natural selection combined with elitist selection, arithmetical one-point crossover, in which a new chromosome is a linear combination of two vectors and non-uniform mutation.

It can be concluded that application of actor model frameworks offers many facilities which provide abstraction layer for low-level operations such as: queuing messages, multithreading calculations, synchronization mechanisms, and location transparency.

Numerical simulations. During simulations optimal braking torques have been calculated using classical genetic algorithm (CGA) and its modification using actor model approach (AMGA). The lane change manoeuvre has been analysed whilst cornering of the articulated vehicle. When the appropriate braking torques not applied the rollover of the vehicle occurs. Additionally, when only the articulated vehicle cornering manoeuvre is considered the truck is stable. Lane change manoeuvre starts at $t=3$ [s] of simulation and in the same time additional braking torques have been applied.These torques act till the end of the simulation ($t_e=6$ [s]). It has been assumed that vehicle initial velocity $v_0=45$ [km/h]. Physical parameters of the articulated vehicle have been taken from [3]. Interpolation of the braking torque has been performed for $m=7$ interpolation nodes which are decisive variables in considered problem. 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. 9).

Fig. 10 shows steering angle course of the articulated vehicle front wheels applied during simulations.

Figure 9. Numbering of the articulated vehicle wheels

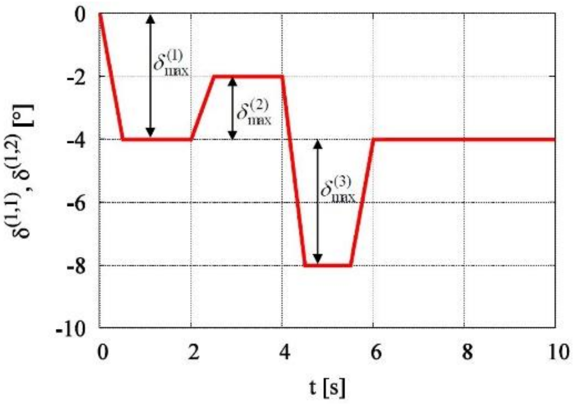


Figure 10. Course of steering angle of the articulated vehicle front wheels

Bulrish-Stoer-Deuflhard [24] method with adaptive step size has been used for integration equations of motion. Computing has been performed on two types of computational unit with OS X macOS Sierra:

– Intel Core i5 CPU 2,60 GHz and 8 GB RAM,
– Intel Core i7 CPU 2,80 GHz and 16 GB RAM.

In the considered problem has been assumed that crossover probability was 0.7 and mutation probability was 0.2. Values obtained for various number of individuals in the population using classical genetic algorithm and genetic algorithm with actor model approach have been presented in tables 2–5. For each variant of calculation assuming a certain number of individuals in a population 10 calculations (cycle) were made, in which the appropriate braking torques wheels of the vehicle starting from a random point were selected. One cycle of calculation was from of a few minutes to about 5 hours for CGA and architecture i5. After calculations, the average values of the objective function and the optimisation time were calculated. The tables show the standard deviation and both the worst, and the best solution obtained in each cycle of calculation. For the results obtained from calculations using the Actor Model, the values designating the gained time (in seconds and percentage) with respect to the classic genetic algorithm by the formulas were specified:

– calculation profit [s]

$$CP_i^{(1)} = CGA_i - AMGA_i\,; \tag{12}$$

– calculation profit [%]

$$CP_i^{(2)} = \frac{CGA_i - AMGA_i}{CGA_i} \cdot 100\%\,, \tag{13}$$

where CGA_i – the computing time achieved for the i-th variant of the calculation obtained by the Classical Genetic Algorithm,

$AMGA_i$ – the computing time achieved for the i-th variant of the calculation obtained by the Genetic Algorithm with Actor Model approach.

Table 2

Optimisation results for CGA for i5 architecture

Number of Mindividuals	Average objective function value	Average calculation time [s]	Number of objective function calls	Standard deviation	Minimal value	Maximal value
10	44.3527	183.4452	300	1.4081	42.5517	47.3425
20	42.9561	374.9039	600	0.3203	42.5268	43.4181
30	42.3353	540.7115	900	0.7821	41.0786	43.4861
40	41.8774	746.6131	1200	0.3637	41.2742	42.6653
50	42.0109	879.6516	1500	0.4859	41.3698	42.8876
60	41.7259	1052.8306	1800	0.4720	40.8842	42.5850
70	41.7031	1275.4891	2100	0.4230	40.9145	42.3244
80	41.5936	1405.3403	2400	0.3300	41.2024	42.1782
90	41.6722	1626.2684	2700	0.2655	41.3342	42.2636
100	41.3054	1792.5185	3000	0.2290	40.9838	41.7496

Table 3

Optimisation results for AMGA for i5 architecture

Number of individuals	Average objective functionvalue	Average calculation time [s]	Number of objectie function calls	Standard deviation	Minimal value	Maximal value	Calc.Profit [s]	Calc.profit [%]
10	43.0768	87.2700	300	0.6098	41.7015	43.8216	96.1752	52
20	42.4478	184.7516	600	0.7222	41.5491	43.4249	190.1523	51
30	42.0810	233.1903	900	0.5624	41.1170	42.8269	307.5211	57
40	41.5456	317.0532	1200	0.3703	41.1456	42.4319	429.5599	58
50	41.5172	406.0199	1500	0.4784	41.0106	42.6789	473.6316	54
60	41.4699	467.7402	1800	0.5597	40.7146	42.6218	585.0904	56
70	41.3832	558.3185	2100	0.3398	41.0061	42.0385	717.1705	56
80	41.2258	626.3760	2400	0.3024	40.8191	41.7889	778.9642	55
90	41.3191	705.6834	2700	0.2363	41.0011	41.6825	920.5849	57
100	41.1643	785.7139	3000	0.2970	40.8084	41.5820	1006.8045	56

Table 4

Optimisation results for CGA for i7 architecture

Number of individuals	Average objective functionvalue	Average calculation time [s]	Number of objectie function calls	Standard deviation	Minimal value	Maximal value
10	44.4153	161.9780	300	1.4819	43.1564	47.3425
20	42.9471	299.7050	600	0.6260	41.5406	43.8861
30	42.4678	427.9175	900	0.7475	41.1176	43.6103
40	42.0691	592.3004	1200	0.2823	41.5196	42.4648
50	41.8844	700.3289	1500	0.4825	41.0427	42.7588

60	41.7983	886.2013	1800	0.3678	41.2749	42.5179
70	41.5894	996.5043	2100	0.4087	41.0990	42.5441
80	41.8207	1161.0532	2400	0.1881	41.6528	42.2099
90	41.6495	1369.3866	2700	0.3453	41.1859	42.2444
100	41.5920	1392.8656	3000	0.2688	41.1542	41.9376

Table 5

Optimisation results for AMGA for i7 architecture

N	Average objective functionvalue	Average calculation time [s]	Number of objectie function calls	Standard deviation	Minimal value	Maximal value	Calc.Profit [s]	Calc. [%]
10	42.7633	88.7395	300	1.0677	40.9899	44.3079	73.2384	45
20	42.0452	86.9445	600	0.7818	41.1425	43.7166	212.7605	71
30	41.4261	123.5882	900	0.3319	40.9504	42.0568	304.3294	71
40	41.5242	166.0152	1200	0.2557	41.1745	41.9267	426.2852	72
50	41.4366	205.9717	1500	0.3590	40.9021	41.9796	494.3573	71
60	41.4389	241.7936	1800	0.4791	40.8295	42.2824	644.4077	73
70	41.2965	292.6774	2100	0.4129	40.8255	41.8980	703.8268	71
80	41.3457	345.6884	2400	0.3955	40.7044	42.0930	815.3647	70
90	41.1608	367.4569	2700	0.2918	40.7503	41.6227	1001.9300	73
100	41.0763	398.2602	3000	0.5143	40.1835	42.0410	994.6054	71

Standard deviation value and the maximum value show that populations consisting of less than 40 individuals in the present case are not suitable for technical applications. In this case, the obtained results have a high randomness and short computing times do not compensate the wide range of values. The results also show that the time gained for the calculations carried out by means of parallel computing using a genetic algorithm with the Actor Model approach significantly accelerate the optimising calculations. More than 50% of the time was gained for architecture i5 and around 70% of speeding up calculations for architecture i7 was reached.

Objective function values obtained for various numbers of individuals in the population using CGA and AMGA have been presented in fig. 11.

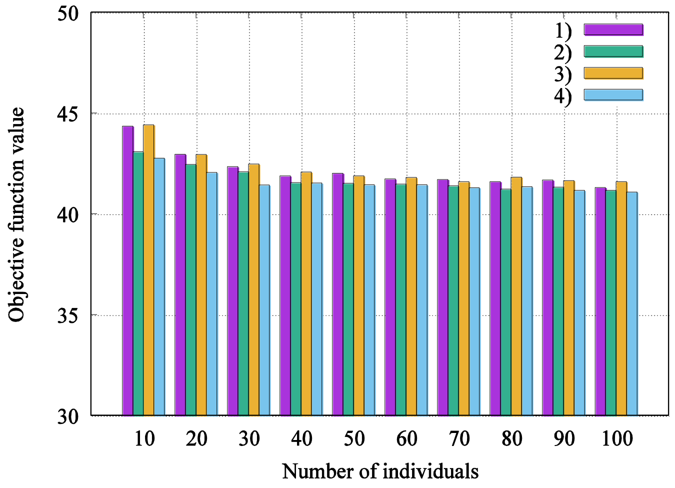


Figure 11. Values of the objective function obtained for various number of individuals for: 1) i5 CGA, 2) i5 AMGA, 3) i7 CGA, 4) i7 AMGA

As shown in fig. 11 the differences between objective function for various numbers of individuals are small, taking into account populations containing from 40 to 100 individuals. Differences between time of optimisation calculations obtained for classical genetic algorithm and its modification with actor model can be seen in fig. 12.

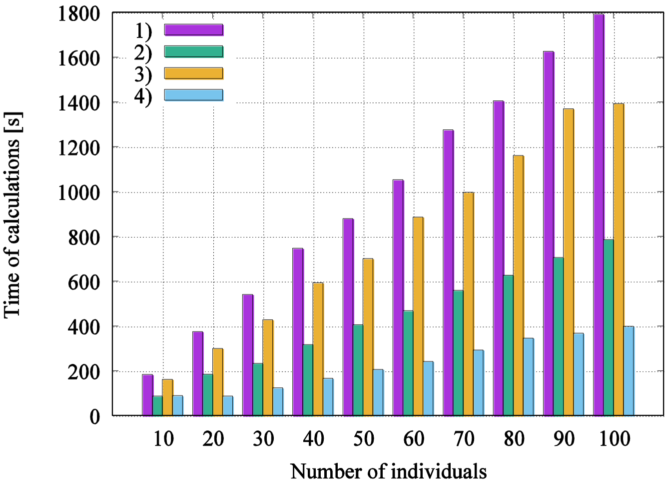
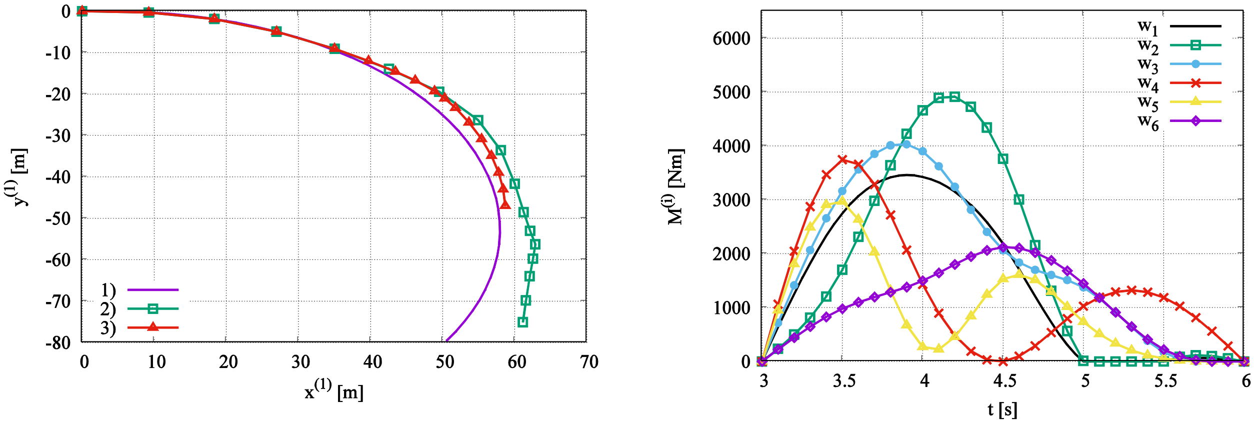


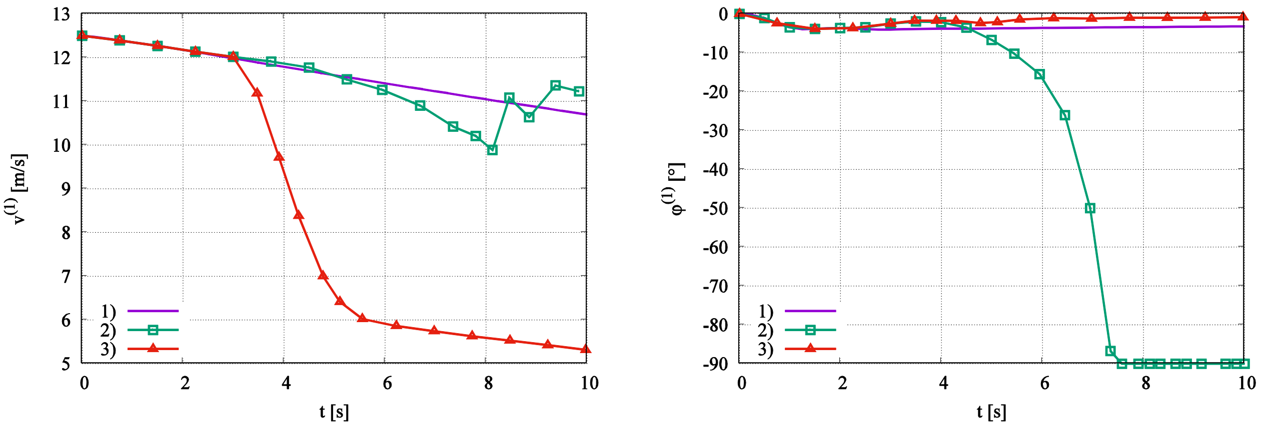
Figure 12. Optimisation calculations time obtained for various number of individuals for: 1) i5 CGA, 2) i5 AMGA, 3) i7 CGA, 4) i7 AMGA

The results shown in fig. 12 prove the enormous saving of computing optimisation time when using AMGA with respect to the classical method. Forty individuals in the population seem to be a good compromise between time and accuracy of calculations. Results obtained for different architectures for the best value of objective function using analysed in the paper types of genetic algorithm are presented in fig. 13–16. Presented courses of the optimal braking torques acting on the wheels $w_1,\ldots,w_4$ are shown in fig. 13–16 a. Other charts (b, c, d) represent courses of the tractor trajectory, its roll angle, and velocity. Figures present also results of simulations for lane following (cornering) manoeuvre 1) and those with the lane change manoeuvre before optimisation 2) and after optimisation 3).



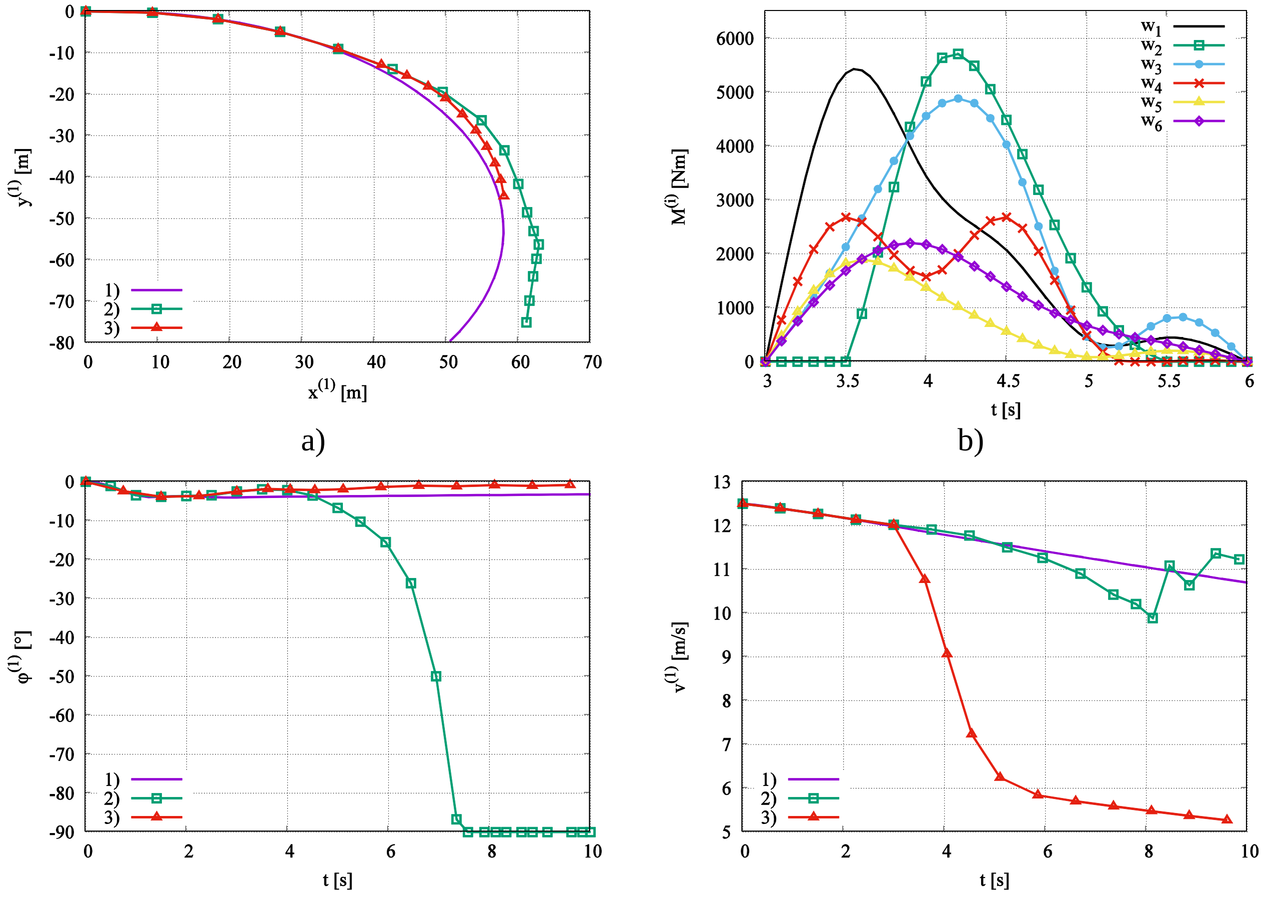
a) b)

Figure 13. Courses of: optimal braking torques a), the tractor trajectory b)



c) d)

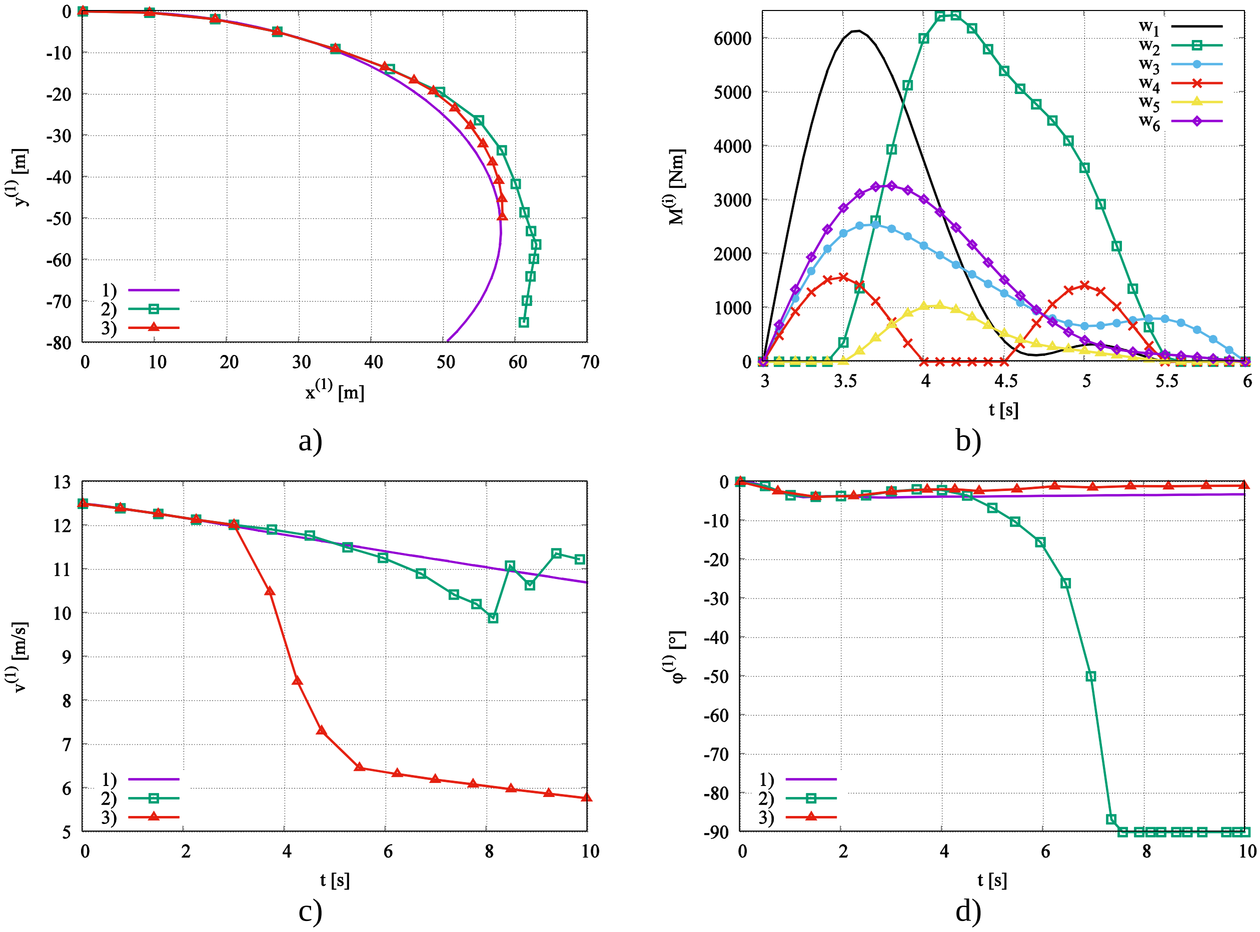
Figure 13. Courses of: optimal braking torques a), the tractor trajectory b), the tractor roll angle c), the tractor total velocity d) for CGA and i5 architecture



a) b)

c) d)

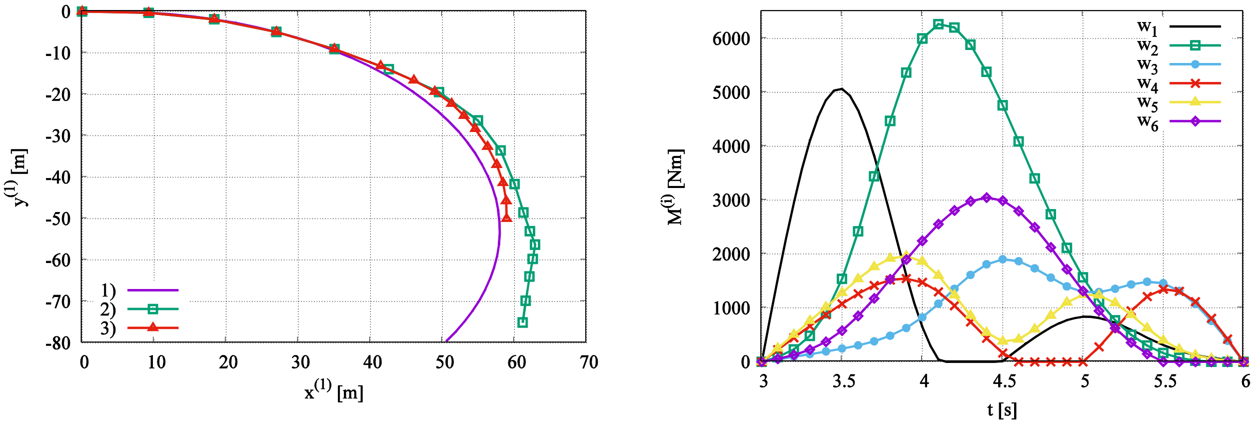
Figure 14. Courses of: optimal braking torques a), the tractor trajectory b), the tractor roll angle c), the tractor total velocity d) for CGA and i7 architecture



a) b)

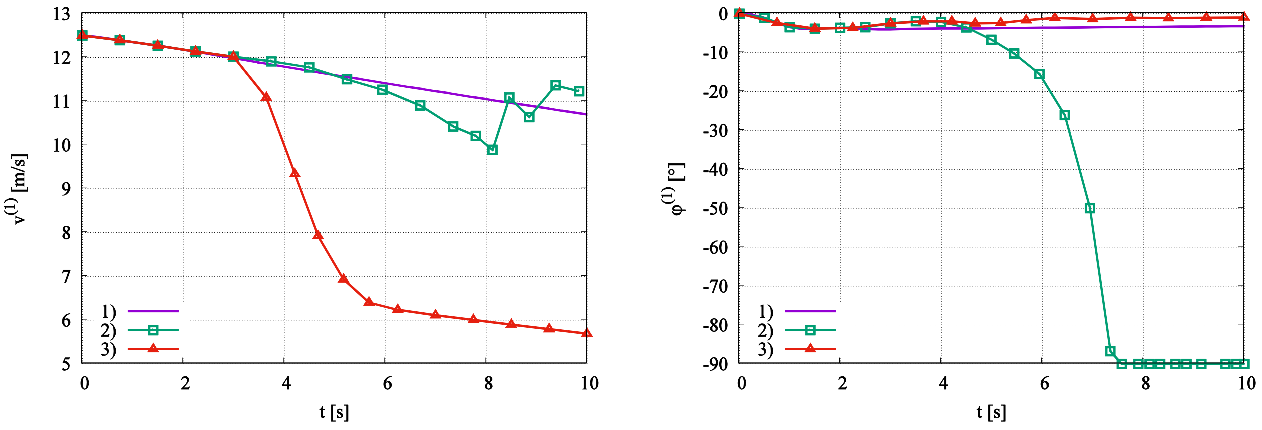
c) d)

Figure 15. Courses of: optimal braking torques a), the tractor trajectory b), the tractor roll angle c), the tractor total velocity d) for AMGA and i5 architecture



a) b)

Figure 16. Courses of: optimal braking torques a), the tractor trajectory b)



c) d)

Figure 16. Courses of: optimal braking torques a), the tractor trajectory b), the tractor roll angle c), the tractor total velocity d) for AMGA and i7 architecture

Large differences in courses of braking torques can be noticed. However, in each case their acting causes the restoration of the stability of a the articulated vehicle. The smallest possible decline of the speed of the tractor was identified for calculations with using the Actor Model approach for i5 architecture (about 55 %).

Conclusions. The articulated vehicle rollover is strongly associated with severe injury and fatalities in highway accidents. Stability can be achieved by the appropriate control of braking torques. In the paper, the problem of controlling brakes has been formulated as the dynamic optimisation task. This task has been solved using classical genetic algorithm and its modification with using the actor model approach. The long duration of the optimisation process results from the necessity of integrating equations of motion in each step. Actor model approach allows reducing time of calculations by dividing computational effort into smaller tasks which are performed by single actor in asynchronousa and parallel way. Results shows that time of calculations obtained for genetic algorithm with various number of individuals using actor model approach is averagely 50 % shorter for i5 and about 70 % shorter for i7 architecture than the time obtained without this modification. It should be noticed that the results have been obtained on a personal computer with a 4th and 8th cores. According to the author, better results can be obtained on servers with the larger number of cores or on cluster. Although the actor model approach is known since the 70s of the last century, but now it can be noticed that interest in this approach in application to modern business systems has been increased. This approach can be easily applied to scientific/numerical applications. In addition to the benefits mentioned previously, the clear source code which is logically split into small atomic parts, well-designed object-oriented architectures, and is easy to maintain and extend, can be obtained.

REFERENCES

1. Eberhard P., Dignath F., Kübler L.: Parallel evolutionary optimization of multibody systems with application to railway dynamics, Multibody System Dynamics, 9 (2), pp. 143–164, (2003).
2. Augustynek K., Warwas K., Polański A.: Application of the genetic algorithms and distributed computing in task of the reduction of vibrations of a satellite, 7th Conference Computer Methods and Systems, pp. 237–242, (2009).
3. Warwas K.: Analysis and control of motion of articulated vehicles with flexible elements, PhD Thesis, University of Bielsko-Biała, Bielsko-Biała, (2008).

4. Plotnikova N. P., Fedosin S. A., Teslya V. V.: Gravitation Search Training Algorithm for Asynchronous Distributed Multilayer Perceptron Model, New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering, Vol. 312, Lecture Notes in Electrical Engineering, pp 417–423, (2015).
5. Umbarkar A.J., Joshi M.S., Hong W.-C., Multithreaded Parallel Dual Population Genetic Algorithm (MPDPGA) for unconstrained function optimizations on multi-core system, Applied Mathematics and Computation, Volume 243, pp. 936–949 (2014).
6. Amaral M., Polo J., Carrera D., Mohomed I., Unuvar M., Steinder M.: Performance Evaluation of Microservices Architectures Using Containers, Network Computing and Applications (NCA), 2015 IEEE 14th International Symposium, pp. 27–34, (2015).
7. The Reactive Manifesto, http://www.reactivemanifesto.org, (2016).
8. Goodwin J., Learning Akka, Pack Publishing (2015).
9. Madeyski L., Jarzębowicz A.: Software Engineering needs Agile Experimentation: A new practice and supporting tool, Advances in Inteligent System and Computing 504, Springer, pp. 149–162, (2016).
10. Ivanovic D., Carro M., Transforming service into cloud-friendly actor networks, Service-Oriented Computing, Springer, pp. 291–305 (2014).
11. Vaughn V.: Reactive Messaging Patterns with the Actor Model Applications and Integration in Scala and Akka, Addison-Wesley (2016).
12. Hewitt C.: Actor Model of Computation for Scalable Robust Information Systems. Inconsistency Robustness, (2015).
13. Lim,Y. H., Tana J., Abramson D.: Solving Optimization Problems in Nimrod/OK using a Genetic Algorithm, Procedia Computer Science, 9, pp. 1647–1656, (2012).
14. Charousset D., Schmidt T.C., Hiesgen R.: CAF – The C++ Actor Framework for Scalable and Resource-efficient Applications, Proc. of the 5th ACM SIGPLAN Conf. on Systems Programming and Applications (SPLASH '14) Workshop AGERE!, New York (2013).
15. Charousset D., Hiesgen R., Schmidt T.: Revisiting Actor Programming in C++, In: Computer Languages, Systems & Structures, Volume 56, pp.105–131, (2016).
16. 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).
17. Yao Z. et al: Dynamic simulation for the rollover stability performances of articulated vehicles, Journal of Automobile Engineering pp. 771–783 (2014).
18. 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).
19. 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, pp. 232–237 (2015).

К. Варвас

МЕТОД ПАРАЛЛЕЛЬНОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА С МОДЕЛЬЮ АКТОРОВ ДЛЯ СОХРАНЕНИЯ СТАБИЛЬНОСТИ СОЧЛЕНЕННОГО ТРАНСПОРТНОГО СРЕДСТВА

В статье исследовано применение модели акторов для управления тормозным моментом колес сочлененного транспортного средства при возникновении опрокидывающего маневра. Для расчета соответствующих тормозных моментов, восстанавливающих стабильность каждого колеса, была использована численная модель сочлененного транспортного средства и проведена ее динамическая оптимизация. Для проведения оптимизации необходимо получить уравнения движения, которые должны быть решены на каждом шаге оптимизации, что занимает много времени. Поэтому для ускорения вычислений предложены параллельные вычисления с использованием модели акторов системы. Модель акторов системы была реализована в генетическом алгоритме. В статье исследован генетический алгоритм с системой акторов и получены результаты динамической *оптимизации.*

Ключевые слова: *параллельные вычисления, модель акторов, генетический алгоритм, оптимизация, сочлененное транспортное средство.*

К. Варвас

МЕТОД ПАРАЛЕЛЬНОГО ГЕНЕТИЧНОГО АЛГОРИТМУ З МОДЕЛЛЮ АКТОРІВ ДЛЯ ЗБЕРЕЖЕННЯ СТАБІЛЬНОСТІ ЗЧЛЕНОВАНОГО ТРАНСПОРТНОГО ЗАСОБУ

У статті досліджено застосування моделі акторів для керування гальмівним моментом коліс зчленованого транспортного засобу в разі виникнення перекидального маневру. Для розрахунку відповідних гальмівних моментів, які відновлюють стабільність кожного колеса, використано чисельну модель зчленованого транспортного засобу й проведено її динамічну оптимізацію. Для проведення оптимізації необхідно отримати рівняння руху, які має бути вирішено на кожному кроці оптимізації, що потребує багато часу. Тому для прискорення обчислень запропоновано паралельні обчислення з використанням моделі акторів системи. Модель акторів системи реалізовано в генетичному алгоритмі. У статті досліджено генетичний алгоритм з системою акторів й отримано результати динамічної оптимізації.

Ключові слова: *паралельні обчислення, модель акторів, генетичний алгоритм, оптимізація, зчленований транспортний засіб.*

УДК 629.5.016

Шапран Ю. Є., Трофименко І. В.

МЕТОД СТАТИСТИЧНОГО ОЦІНЮВАННЯ ТА ПРОГНОЗУВАННЯ ПАРАМЕТРА ПОТОКУ ВІДМОВ АГРЕГАТІВ СИСТЕМИ АВТОМАТИЧНОГО КЕРУВАННЯ СУДНОВИХ КОМПЛЕКСІВ ЗА ДАНИМИ ЕКСПЛУАТАЦІЙНИХ СПОСТЕРЕЖЕНЬ

Запропонований підхід статистичного оцінювання й прогнозування параметра потоку відмов агрегатів системи автоматичного керування судновими комплексами за даними експлуатаційних спостережень дає можливість сформувати керування технічним станом агрегатів суднових комплексів, що є суттєвим під час вирішення завдання забезпечення заданого рівня надійності складних систем такого типу.

Ключові слова: *статистичне оцінювання, прогнозування параметра потоку відмов, системи автоматичного керування, експлуатаційні спостереження.*

Актуальність. Аналіз засвідчив, що в сучасних умовах розвитку морського флоту одним із найважливіших напрямків забезпечення заданого рівня безпеки є автоматизація керування судновими комплексами. Під час розроблення та проектування автоматичних систем керування судновими комплексами виникає необхідність розроблення методу статистичного оцінювання й прогнозування параметра потоку відмов агрегатів системи автоматичного керування судновими комплексами за даними експлуатаційних спостережень.