Minimization of real-time factor in “miMA” vehicle-driver-road model

M Maniowski

Institute of Automobiles and Internal Combustion Engines,
Cracow University of Technology, Poland
E-mail: mmaniowski@pk.edu.pl

Abstract. The paper concerns further improvements at numerical level of computational time and accuracy of formulated by the Author model of vehicle-driver-road (called “miMA”) for motorsport optimization applications. Selection of the best ordinary differential equations (ODE) solver (i.e.: ode23t, ode23s, ode15s, ode45) and its parameters (i.e.: sampling frequency, relative tolerance, absolute tolerance, maximum step) is carried out based on the formulated optimization algorithm, instead of trial and error methods. After optimization, it turned out, that the initial real-time factor (RTF = 0.20 for $t_{sim} = 1.6$ s by using ode15s solver) of “miMA” model, can be reduced by even 50% (to RTF = 0.087 for $t_{sim} = 0.695$ s by using ode23t solver) without scarifying too much the model numerical accuracy. These are key features for effective solving of large optimization tasks, like a lap time minimization with full vehicle model.

1. Introduction
The paper concerns some further improvements of an innovative method, formulated by the Author in Matlab environment and called “miMA”, for optimization in motorsport applications. “miMA” approach is characterized by:

• multibody spatial model of a vehicle with hundreds of discrete parameters specialized in motorsport, like off-road [1], racing [2, 3], rally [4], drifting [5] or karting;
• effective code yielding proper balance between model accuracy and computation time;
• combined optimization of driver actions for closed-loop manoeuvres, vehicle chassis and motion trajectory;
• implemented genetic algorithm which enable global search of nonlinear and multi-criteria tasks;
• substitution of actual driver actions by additional optimization variables, what emulates driver adaptation without using oversimplified driver models.
Figure 1. „miMA” model of driver-vehicle-road system used for optimisation tasks in motorsport.

In order to solve this kind of large optimization problems in reasonable time, a proper balance between model accuracy and its computation time, needed for the goal functions and constraints evaluations, is essentially required.

For example [3], let’s consider optimization of Ford Focus ST170 (Front Wheel Drive, 170 horsepower engine, 1250 kg curb weight with 60/40% distribution, limited slip differential and semi-slick tires) prepared for a track racing, which is negotiating a selected track part with RH corner in racing pace (figure 2). Formulated „miMA” simulation model of driver-vehicle-road system with 26 generalized coordinates and about 450 parameters, after positive experimental verification has been utilized for optimization by using genetic algorithms. The combined optimization includes 28 decision variables, with 20 variables responsible for the driver commands and 8 variables for the car chassis parameters. Two performance criteria, i.e. section time and exit velocity, are defined as the optimization goals. Different strategies for improving performance of the race driver, FWD car chassis and the motion trajectory can be found based on the optimization results.

In the mentioned case of the nonlinear system optimization with 28 decision variables, 2 goal functions, and 5 constraints, about 30k iterations are needed for the genetic algorithm to finish the optimisation with satisfying results. Those iterations take ultimately about 12 hrs, of time, which of course is depended on an isolated computation time of “miMA” model.
Even for a so detailed vehicle model it was managed to significantly reduce the model computational burden. Simulation of this single road scenario (figure 2) of 8 s duration time takes about 1.6 s (to solve an initial value problem describing the considered vehicle dynamics by utilizing initially chosen ode15s algorithm in Matlab ver. 9 [6] on Laptop CPU 3GHz, 8GB RAM). This yields the real-time factor (defined as ratio of simulation time and real manoeuvre time) equal RTF = 0.20. Hence, the model can be computed about five times faster, than the real time of the considered manoeuvre.

This beneficial RTF of “miMA” model was achieved mainly through (at structure level):

- selection of the model mechanical structure with a medium numerical stiffness, describing only a gross motion dynamics (up to 20 Hz);
- formulation of the model motion equations as ordinary differential equations (ODE) in a minimal form by applying the Newton-Euler approach and solving the wheel suspension kineto-statics problem in a closed form (in that way, inefficient formalism with differential-algebraic-equations is not involved);
- avoiding model components with discontinues characteristics (like stick-slip friction);
- avoiding model components with control units with high gains (like Anti-Block-System).

In order to further minimize (at numerical level) “miMA” model computational cost and the model deviation, selection of the best ODE solvers and their parameters is carried out based on the following optimization algorithm. This kind of multivariate and highly nonlinear problems are hard to solve effectively by using trial an error methods only.

2. Minimization of simulation time and model deviation

Selection of the best ODE solver (in Matlab) and its parameters for given tasks is carried out as an optimization problem. The optimization goal is to minimize the following criteria:

$\min w = [t_{\text{sim}} \Delta x]_{1x2}$                                    \hspace{1cm} (1)

where:

- $t_{\text{sim}}$ – measure of computation time [s] (the lower, the better);
- $\Delta x$ – measure of computation deviation ($\Delta x = \text{norm}(X - X0)$) (lower-better);
- $p_{\text{ode}}$ – ODE solver parameters (defined in table 1).

Usually, the defined criteria (Eq.1) are contradictory (time and accuracy). Then, a compromising solution can be chosen.

Figure 2. “miMA” model of Focus ST170 with 26 generalized coordinates.
Table 1. Definition of ODE solver parameters (d) utilized for optimization.

| No | Description                  | Initial value | Range         | Normed range (d) |
|----|------------------------------|---------------|---------------|------------------|
| 1  | Sampling frequency [Hz]      | 40            | 20÷60         | -1÷1             |
| 2  | Relative tolerance [-]       | 1e-03         | 1e-02÷1e-04   | -1÷1             |
| 3  | Absolute tolerance [-]       | 1e-03         | 1e-02÷1e-04   | -1÷1             |
| 4  | Maximum step [s]             | 1e-03         | 1e-02÷1e-04   | -1÷1             |

In order to minimize this base line real-time factor (i.e. RTF = 0.20 for $t_{sim} = 1.6$ s by using ode15s solver), firstly, different ODE solvers are considered.

3. Influence of different ODE solvers
   a) Algorithm ode15s (Matlab)
   The first tested algorithm ode15s is a variable-step, variable-order (VSVO) solver based on the numerical differentiation formulas (NDFs) of orders 1 to 5. Optionally, it can use the backward differentiation formulas (BDFs, also known as Gear's method) that are usually less efficient. It is recommended to use ode15s when ode45 fails or is very inefficient and you suspect that the problem is stiff, or when solving a differential-algebraic equation (DAE) [7].

   The obtained results of the minimization problem (Eq.1) by using pattern search method [6], are presented in figure 3. The starting value of “miMA” model simulation time ($t_{sim}=1.1$ s) is consistently minimized by the algorithm to vary promising level of 0.66 s (nearly 50% reduction!), after 55 iterations. The final set of the normed decision variables (Eq.1), is the following $d = [-1 \ 0.8 \ -0.5 \ 0]$. It can be noticed, that the 4th variable seems to be sub-optimal.

![Figure 3](image-url)
b) Algorithm ode45 (Matlab)
The following tested algorithm is ode45. Most of the time it should be the first solver you try [6]. The ode45 is based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair. It is a single-step solver, in computing it needs only the solution at the immediately preceding time point [7]. This algorithm belongs the medium order methods, that can solve numerically nonstiff differential equations.

The obtained results of the minimization problem (Eq.1) by using pattern search method [6], are presented in figure 4. The starting value of “miMA” model simulation time \( t_{\text{sim}} = 2.3 \) s is minimized by the algorithm to a not sufficient level of 1.94 s (nearly 30% reduction), after 30 iterations. The final set of the normed decision variables (Eq.1), is the following \( \delta = [-0.4 \quad -1 \quad -0.1 \quad -0.02] \). It can be noticed, that those changes of the decision variables not correspond with the one obtained in figure 3. Again, the 4th variable seems to be sub-optimal.

![Figure 4. Minimization results of computation time \( t_{\text{sim}} \) for ode45 algorithm.](image)

C) Algorithm ode23s (Matlab)
The ode23s is based on a modified Rosenbrock formula of order 2. Because it is a single-step solver, it may be more efficient than ode15s at solving problems that permit crude tolerances or problems with solutions that change rapidly. It can solve some kinds of numerically stiff problems for which ode15s is not effective. The ode23s solver evaluates the Jacobian during each step of the integration, so supplying it with the Jacobian matrix is critical to its reliability and efficiency [7].

The obtained results of the minimization problem (Eq.1) by using pattern search method [6], are presented in figure 5. The starting value of “miMA” model simulation time \( t_{\text{sim}} = 13 \) s is minimized by the algorithm to the worst level of 7.70 s, after 20 iterations. The final set of the normed decision variables (Eq.1), is the following \( \delta = [-0.3 \quad 1 \quad -1 \quad 1] \).
Figure 5. Minimization results of computation time ($t_{sim}$) for ode23s algorithm.

d) Algorithm ode23t (Matlab)

The last tested algorithm is ode23t, which implements the trapezoidal rule using a “free” interpolant. This solver is preferred over ode15s if the problem is only moderately stiff and you need a solution without numerical damping. ode23t also can solve differential algebraic equations (DAEs) [7].

The obtained results of the minimization problem (Eq.1) by using pattern search method [6], are presented in figure 6. The starting value of “miMA” model simulation time ($t_{sim} = 0.95$ s) is consistently minimized by the algorithm to so far the best result of 0.64 s (nearly 30% reduction), after 65 iterations. The final set of the normed decision variables (Eq.1), is the following $\vec{d} = [-0.9, 0.8, -0.4, -1]$. It can be noticed, that those changes of the decision variables correspond well (except of 4th) with the one obtained in figure 3.
4. Influence of different solver \textit{ode23t} parameters

In the following part, the algorithm \textit{ode23t}, selected as the best in Chap.3, is further analysed with respect to the significance level of all the decision variables (table 1).

The obtained results of the minimization problem (Eq.1) with only 2 decision variables (ie. Relative tolerance and Absolute tolerance, in tab.1) chosen, are presented in figure 7. The starting value of “miMA” model simulation time ($t_{\text{sim}} = 0.9$ s) is consistently minimized by the algorithm to a level of 0.65 s, after 20 iterations. The final set of the normed decision variables, is the following $\mathbf{d} = [0.9, -0.8]$. The obtained results confirm, that the chosen variables are the most significant, because the minimized simulation time is roughly the same like in figure 3.
Figure 7. Minimization results of computation time ($t_{\text{sim}}$) for $\texttt{ode23t}$ algorithm for only 2 decision variables (ie. Relative tolerance and Absolute tolerance).

5. Influence of compromise between computation time and model deviation

In the last part, selected as the best (Chap.3) the algorithm $\texttt{ode23t}$ with only 2 the most significant variables (Chap.4), is further analysed with respect to the compromise between simulation time and numerical deviation (Eq.1).

The obtained results of the minimization problem (Eq.1) with 2 objectives, are presented in figure 8. This multi-objective problem is solved by using Genetic Algorithm with nondominated sorting. The obtained Pareto front with a convex profile (figure 8), confirms trade-off character between the defined criteria. It means, the minimal time and the minimal model deviation cannot be achieved at the same time.

In the case of “miMA” application for optimization tasks, a compromising solution is chosen (noted in figure 8), because it yields still highly reduced simulation time ($t_{\text{sim}} = 0.695$ s) without scarifying too much on the algorithm accuracy ($\Delta x = 32$). The shortest time solution ($t_{\text{sim}} = 0.69$ s) seems to be not meaningful due to a huge increase of the model deviation ($\Delta x = 116$).
Figure 8. Pareto optimal results for ode23t algorithm evaluated by objective 1 (computation time: \( t_{\text{sim}} \)) and objective 2 (model deviation: \( \Delta x \)).

6. Concluding remarks and further steps
After optimization of different ODE solvers and their parameters, it turned out, that the initial real-time factor (RTF = 0.20 for \( t_{\text{sim}} = 1.6 \) s by using ode15s solver) of “miMA” model, can be reduced by even 50% (to RTF = 0.087 for \( t_{\text{sim}} = 0.695 \) s by using ode23t solver on the same computation platform) without scarifying too much the numerical algorithm accuracy (figure 8).

The obtained computation time of formulated the Author “miMA” vehicle-driver-road model, with 26 DOFs and about 450 parameters, is nearly 10 times shorter than the real time. Therefore, 30k of iterations needed for the genetic algorithm to finish the combined optimisation [3] with satisfying results, currently take about 6 hrs of time (instead of 12 hrs).

Comparing this performance feature with a vehicle model (with a similar structure) formulated by using a commercial multi-body-simulation software (like ADAMS/Car), the computational time would be increased in this case even 50 times [8], what significantly impedes the effective solving of large optimization tasks.

References
[1] Maniowski M 2014 Optimization of wheel suspension dampers in off-road rally car Proceedings of the Institute of Vehicles 4 (100) pp. 57-69
[2] Maniowski M 2016 Optimization of driver actions in RWD race car including tyre thermodynamics Vehicle System Dynamics 54 4
[3] Maniowski M 2016 Optimization of driver and chassis of FWD racing car for faster cornering The Dynamics of Vehicles on Roads and Tracks, Proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics (IAVSD 2015), Graz, Austria, pp. 77-86
[4] Kluziewicz M, Maniowski M 2016 Calibration procedure of measuring system for vehicle wheel load estimation IOP Conference Series: Materials Science and Engineering 148 1
[5] Maniowski M 2016 Multi-criteria optimization of chassis parameters of Nissan 200 SX for drifting competitions IOP Conference Series: Materials Science and Engineering 148 1
[6] MATLAB – software manual (www.mathworks.com)
[7] Press W H 2007 Numerical Recipes The Art of Scientific Computing 3rd edition Cambridge University Press
[8] Nabagło T, Jurkiewicz A, Kowal J 2015 Semi-Active Suspension System for 2S1 Tracked Platform in Application of Improvement of the Vehicle Body Stability Applied Mechanics and Materials 759