THE EXPERIENCE WITH OPTIMISATION OF HIGHLY NON-LINEAR DYNAMIC SYSTEMS BY GENETIC ALGORITHMS IN MATLAB ENVIRONMENT

Zbynek Šika, Pavel Steinbauer, Michael Valášek, [1]

Abstract: The paper summarises experience with optimisation of large, complicated, non-smooth and non-linear dynamic systems, in particular controlled mechanisms by genetic algorithms. The typical problem and its cost function is described together with reasons for introduction of genetic algorithms. Implementation issues are discussed and results are compared with capabilities of other optimisation methods and packages. The work is based on the ready-to-use, freeware package GAOT.

Design of Complex Mechatronic Systems

Design of mechatronic system is nowadays often based on computer modelling and simulation. The more and more powerful computers allows us to model reality more and more accurately. This means that models of subsystems with different nature (mechanical, electrical, hydraulic etc.) must be interconnected. The most accurate models of some elements are based on the measured data and provided into modelling software as look-up tables, the functions are not smooth, contain hysteresis etc.

It is often more convenient to prepare models of different subsystems in specialised software package (multi-body systems in SIMPACK or ADAMS, FEM systems by ANSYS, control, logic and many others in MATLAB/SIMULINK. These subsystems can then be interconnected on many levels (function call interface where model is interpreted, symbolic code interface which exports compiled code etc.).

Genetic Algorithms and Global Optimisation

The objective function usually has a global character, i.e. there are many local extremes. Unfortunately, traditional optimisation methods are restricted so that on considered interval of parameter only one extreme can exists, require smoothness etc.. In addition, the most powerful mathematical optimisation routines require the first and second derivatives of the objective function. These are not, however, available in many cases due to the nature of simulation model described in previous paragraph.

That's why the genetic algorithms were used besides the global optimisation methods with interesting results. Although they are quite computationally expensive, it provides great robustness and global character of the search, not requiring any preconditions.

The genetic algorithm is a model of machine learning which derives its behaviour from a metaphor of some of the mechanisms of evolution in nature. This is done by the creation of a POPULATION of INDIVIDUALS (in our case the INDIVIDUAL is one set of parameters, one guess), represented by chromosome, a binary, character or real number string that is analogous to the base-4 chromosomes that we see in our own DNA. The individuals in the population then go through a process of simulated "evolution".

In general, the fittest individuals of any population have the best chance to reproduce and survive to the next generation, thus improving successive generations. Thus FITNESS, in the sense of optimisation, is equivalent to the objective function value.

Simple bit manipulation operations allow the implementation of CROSSOVER, MUTATION and other genetic operations.

[1] Department of Mechanics, Faculty of Mech. Engineering, Czech Technical University in Prague

valasek@fsik.cvut.cz, zbynek@valas.fsik.cvut.cz, steinb@fsik.cvut.cz

The genetic algorithm is implemented in a way that involves the following cycle: Evaluate the FITNESS of all of the individuals in the population. Create a new population by performing operations such as crossover, fitness-proportionate REPRODUCTION and mutation on the individuals whose fitness has just been measured. Discard the old population and iterate using the new population.

One iteration of this loop is referred to as a GENERATION. The first generation (generation 0) of this process operates on a population of randomly generated individuals or individuals given by qualified guess. From there on, the genetic operations, in concert with the fitness measure, operate to improve the population.

The Case Study Problem Description

The dynamic interaction between heavy vehicles and road infrastructure (roads or bridges) has recently received increased attention. The static values of road-tyre forces are being regulated by the present standards. However the dynamic part of road-tyre forces causes significantly increased damage of roads and increased loading of bridges. The usage of controllable shock absorbers instead of passive ones can improve the dynamic behaviour of truck suspension. The goal is to decrease road damage as well as to increase ride comfort for a broad range of road irregularities by the suitable choice of control law structure and parameters.

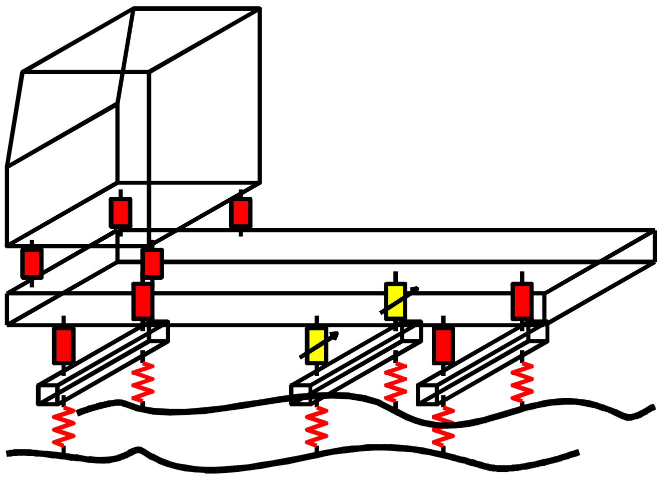

Figure 1 - 3D mechanical scheme of considered truck

The reference simulation model (Figure 1) is the 3D simulation model of the truck prototype considering only the vertical motion of axles and chassis. On the other hand all suspension (air and leaf springs, tyres) and damping elements are treated as non-linear. The multi-body dynamic simulation software SIMPACK has been used for this purpose (Fig.2). The truck is considered fully loaded passing the road with the velocity of 72 km/h. Many of reference inputs (such as stochastic roads, bumps, pots and ramps) have to be taken into account.

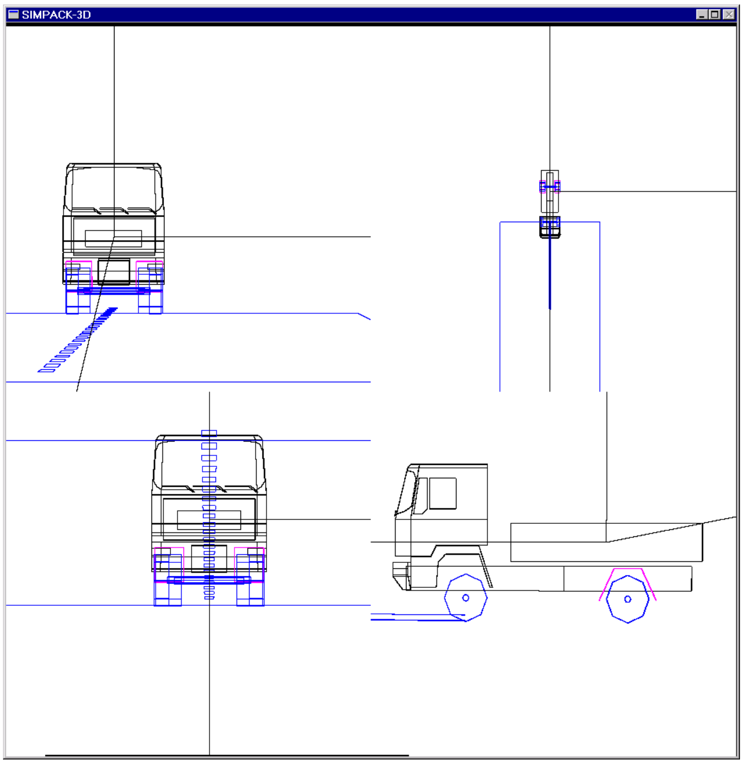

Figure 2 - The SIMPACK truck simulation model - the graphical representation

The evaluation criteria (performance measure) are road friendliness and comfort for driver and load. The Dynamic Load Stress Factor (DLSF) [Hedrick89] is taken as an basic evaluation criterion of the road damage

$$DLSF = 1 + 6DLC^2 + 3DLC^4 \qquad (1)$$

where DLC (Dynamic Load Coefficient) is

$$DLC = \frac{RMS(Dynamic\ Tyre\ Force)}{Static\ Tyre\ Force} \qquad (1a)$$

The objective is to minimise this criterion. Alternatively, the simple integral criterion of the tyre force can be used:

$$F_{\text{int}\,eg} = \int_0^T (F_{actual} - F_{stat})^2\, dt \qquad (2)$$

The ISO weighted acceleration RMS is used for the ride comfort criterion. It is required not to deteriorate it in comparison with passive suspension.

The highly non-linear properties of truck suspension exclude the efficient usage of standard linear control synthesis. The multi-objective parameter optimisation (MOPO) method [Kortüm98] of control synthesis has been chosen as the most suitable for this task.

The basic structure of control law has been proposed based on the physical considerations. An important principle for the ride comfort is the so-called sky-hook developed by Karnopp [Karnopp74]. The novel control concept, so-called ground-hook, as a fictitious damping element between the wheel and the ground parallel with the tyre has been introduced, [Valášek97]. The motivation of this concept is to develop an equivalent to sky-hook for the reduction of dynamic tyre-road forces. The preservation of low accelerations of sprung mass has been reached by the combination of sky-hook and ground-hook and ground-hook extensions.

The basic ground-hook concept has been implemented to the semi-active suspension. The following explanation (Fig.3),(3) is based on the simple linear quarter car model. However, it should be noted that the control law parameter determination is then performed on the non-linear quarter car model and all evaluations are based on the verified non-linear 3D simulation model.

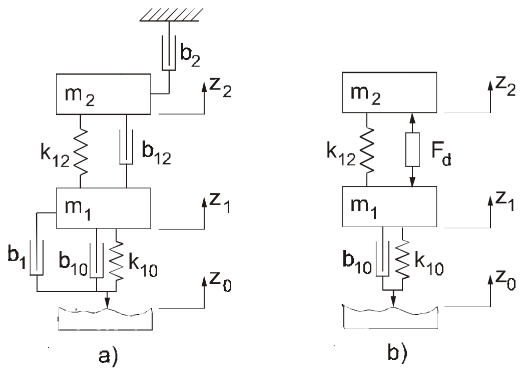

Figure 3 The dynamic scheme of 1/4 of car

The meaning of quantities z_2, z_1 and z_0 as well as control law parameters is obvious from Fig. 3.The control law of EGH (extended ground-hook) in combination with sky-hook is, [Valášek97],

$$F_d = b_1(\dot{z}_1 - \dot{z}_0) - b_2\,\dot{z}_2 - b_{12}(\dot{z}_2 - \dot{z}_1) + \Delta k_{10}(z_1 - z_0) - \Delta k_{12}(z_2 - z_1) \tag{3}$$

The parameters of the extended ground-hook were originally considered constant for the whole shock absorber velocity interval. However, this concept brought some unnecessary limitations to the tuning of the control law. The strong non-linearity of the controlled shock absorber, especially the asymmetry of characteristics, can be taken into account for the determination of control-law parameters. Therefore a non-linear EGH version, which enables the state-dependent coefficients (gains) of the EGH control law (3), was developed. Four subintervals of the relative damper velocity $v = \dot{z}_2 - \dot{z}_1$ have been defined namely the high negative, low negative, low positive and high positive. Their dependence has to be determined by the global optimisation, MOPO approach.

Optimisation Implementation

As mentioned above, the truck simulation model (see Figure 2) was created in the MBS[2] software package SIMPACK[3], which allows us to fully respect non-linearities of the real truck, including authentic tyre modelling, flexible frame and more. Furthermore, the models of truck subsystems are already included in the package.

On the other hand, control model implementation and design can be more conveniently performed in MATLAB/SIMULINK environment. Thus the truck SIMPACK [SIMPACK98] model (including the semi-active

[2] Multi Body Simulation

[3] Intec, GmbH and DLR (Wessling)

dampers) was connected via SIMAT interface [Vaculin98] with SIMULINK (see Figure 4). The simulation was driven by SIMULINK numerical integrator.

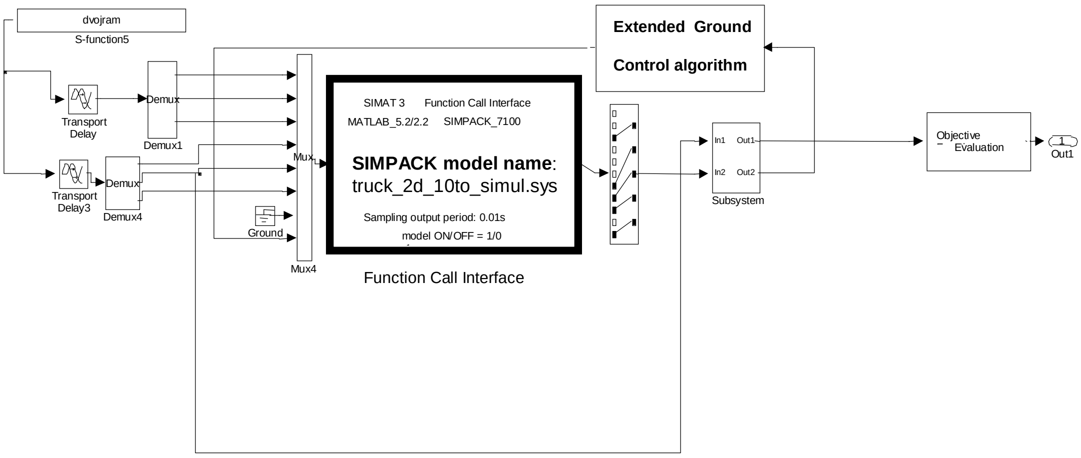

Figure 4 - The SIMULINK model of optimised system

The optimised parameters, i.e. control law coefficients, were defined as MATLAB global variables in SIMULINK block diagram so that they could be modified from MATLAB workspace. The numerical integration was started by command line syntax (sim command) and so included into the objective function evaluation. The output of objective function was directly DLSF value (see equation (1)), the inputs were optimised parameters.

From that point the objective function was ready for optimisation by any optimisation method or package available in MATLAB environment. The genetic optimisation was based on the freeware MATLAB toolbox GAOT [Houck95]. This package provides many sophisticated functions, including functions, which realise the basic genetic operations (i.e. mutation, crossover). On the other hand, it is possible to start with function ga(), which is entry point into the GAOT toolbox and provides quite good performance with minimum setting of genetic algorithm and later modifications are possible. This approach makes package GAOT user friendly and enables newcomer to start work easily. We used the "floating point" mode, which is an extension of classical "binary" representation of individuals and provides better results then traditional binary one. The "fitness" function had to be set-up carefully as the "ga" function is maximising and the usual goal of optimisation routines is minimisation. This can be easily solved by multiplication by -1 of the objective function.

Results and Conclusions

The genetic optimisation provided good performance, showing typically fast convergence at first populations and very slow at the later ones (see Figure 5). The usage for the global optimisation of our objective functions proved to be very efficient. No gradient information is needed and the robust search process of the optimisation parameter space is possible even for the high non-linearities included. The results of the genetic optimisation provided efficiently possible regions of global extremes which is otherwise very difficult to find. These results can be further tuned by local optimisation methods (simplex method *fmins* etc.).

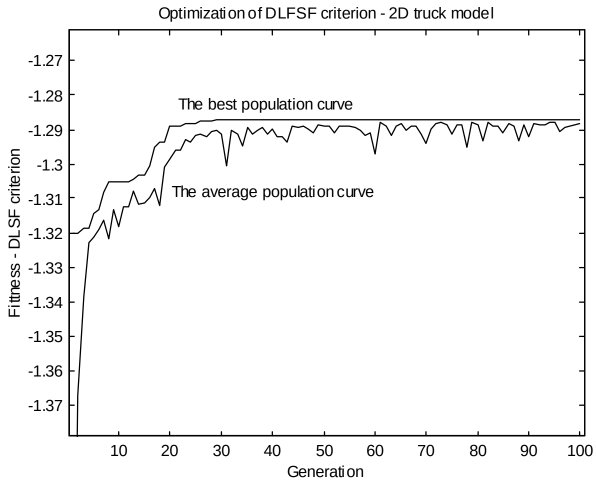

Figure 5 - The typical genetic optimization progress of population DLSF objective function criterion (see Eq. 1)

In the case of mechatronic/(mixed nature) systems modelling the actual subsystems needs to be often modelled by the special simulation tools. The savings of the model design time are often much more important than the savings of computational time itself. The software tool interfacing is more efficient than re-building of models in specific optimisation environment, where in addition these models have to be simplified. This multi-tool approach highly advantages methods which needs only objective function values.

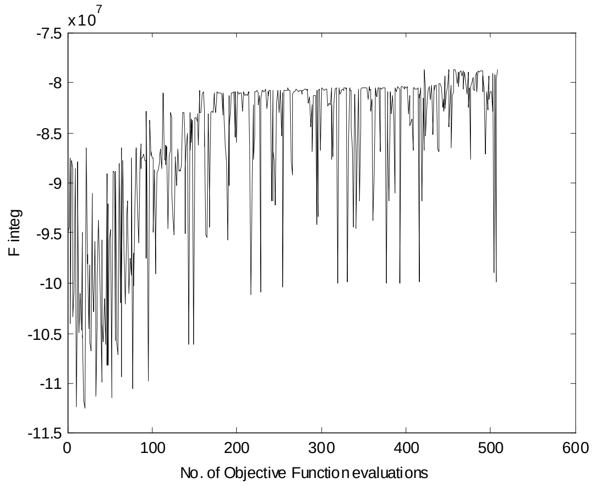

Figure 6 - The dynamic road force integral criterion (see Eq. 2) (all objective function evaluation shown)

Due to its nature, i.e. the whole population of possible extremes is evaluated "at once", are genetic algorithms very suitable for parallel computing, either on multi-processor machines or on clusters of computers. Unfortunately, neither MATLAB nor GAOT support parallel computing yet.

Acknowledgements

The authors appreciate the kind support by the grants MŠMT J04/98:212200008 "Development of methods and tools of integrated mechanical engineering" and MŠMT J04/98:212200003 "Development of algorithms of computational mechanics and their application in engineering".

References

[Vaculin98] Vaculín O., Steinbauer P.: MATLAB/Simulink connection with commercial packages SIMPACK and UFO, seminář MATLAB, SIMULINK, STATEFLOW, Praha 98

[Houck95] Chris Houck, Jeff Joines, and Mike Kay: A Genetic Algorithm for Function Optimization: A Matlab Implementation, NCSU-IE TR 95-09, 1995, **Chyba! Nenalezen zdroj odkazů.**

[Kortüm98] Kortüm, W., Valášek, M., Vaculín, O. (eds.): SADTS Final Technical Report, DLR, Wessling 1998

[Karnopp74] Karnopp, D., Crosby, M.J., Harwood, R.A.: Vibration Control Using Semi-Active Force Generators, Transactions of ASME, J. of Engineering for Industry, 96(1974), pp. 619-626

[Hedrick89] Yi, K., Hedrick, J.K.: Active and Semi-Active Heavy Truck Suspensions to Reduce Pavement Damage, SAE Technical Paper 892486, 1989

[SIMPACK98] SIMPACK - User Manual, German Aerospace Research Institute (DLR), Oberpfaffenhofen 1998

[Valášek97] Valášek, M. et al.: Extended Ground-Hook -- New Concept of Semi-Active Control of Truck's Suspension, VSD, 27(1997), pp. 141-151