

ASPECTS CONCERNING THE MODERN MODELING & SIMULATION TECHNIQUES OF THE MECHANICAL SYSTEMS

Cătălin ALEXANDRU

"Transilvania University of Braşov", Product Design & Robotics Dept., calex@unitbv.ro

Keywords: mechanical system, modeling, simulation, digital mock-up, virtual prototyping.

Abstract: The paper presents specific aspects regarding the modern modeling & simulation techniques of the mechanical systems, such as digital mock-up and functional virtual prototyping. The following specific aspects are taken into consideration: the advantages of the system-focused simulation techniques relative to the traditional ones, the software components of the virtual prototyping platform and the connections between them (i.e. the data transfer), the analysis flow chart of the mechanical systems, and the virtual prototyping phases. Finally, a complex example is presented for demonstrating the capabilities of the virtual prototyping technique, in the field of automotive engineering (the design and simulation an active suspension system).

1. INTRODUCTION

The determination of the "real behavior" has become a priority in the functional design & analysis of the mechanical systems. On the basis of advanced computer programs, the possibility to build models of not just parts but entire systems is assured, and then to simulate their behavior and optimize the design before building an expensive hardware (physical) prototype. The traditional CAD/CAM/CAE techniques were orientated on the concept referred to as "art-to-part", which is directed toward the design, development and manufacturing of the mechanism's components (subsystems). However, optimal component design does not always leads to optimal system design. The interaction of form, fit, function and assembly of all parts in a mechanism is a major contributor to overall product quality. The only way to increase quality, and reduce time and cost, consists now of functional virtual prototyping applied to system level [9].

Various scientific papers reveal a growing interest on analysis methods for "multi-body" systems that allow the self-formulating algorithms, having in view to develop powerful modeling and simulation programs, which facilitate building and simulating a computer (virtual) model of any mechanism [1-6]. These types of programs were lanced in commercial versions even in the 1980's but in the last decade a new type of studies were defined through their use: Virtual Prototyping. This technology consists mainly in conceiving a detailed model and using it in a virtual experiment, in a similar way with the real case. Virtual Prototyping is a software-based engineering process that enables modeling the mechanism, simulating the motion under real operating conditions and, finally, optimizing its behavior. An important advantage of this kind of analysis - simulation consists in the possibility of make virtual measurements in any point and area of the mechanism and for any parameter (displacements, velocities, accelerations, forces etc.). In this way, the design engineers can take decisions on any design changes without going through physical prototype building and testing.

In these terms, the paper presents the following issues regarding the modern modeling & simulation techniques of the mechanical systems: the software components of the virtual prototyping platform and the connections between them (i.e. the data transfer), the analysis flow chart of the mechanical systems, and the virtual prototyping phases. For demonstrating the capabilities of the virtual prototyping technique, the virtual model of an active suspension system is developed & simulated by using specific MBS – Multi-Body Systems (ADAMS) and DFC – Design for Control (EASY5) software solutions.

2. THE VIRTUAL PROTOTYPING PLATFORM

Generally, the virtual prototyping platform includes the following software products (fig. 1): CAD - Computer Aided Design (ex. CATIA, PROENGINEER, SOLIDWORKS); MBS – Multi-Body Systems (ex. ADAMS, SD-EXACT, PLEXUS); FEA – Finite Element Analysis (ex. NASTRAN/PATRAN, ABAQUS, ANSYS); DFC – Design for Control (ex. MATLAB/Simulink, EASY5, MATRIXx). The MBS software is the main component of the virtual prototyping platform, allowing analyzing, optimizing, and simulating the mechanical system under real operating conditions.

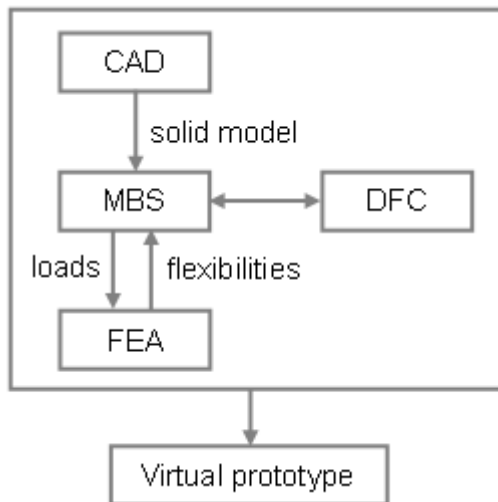


Fig. 1. Virtual prototyping platform

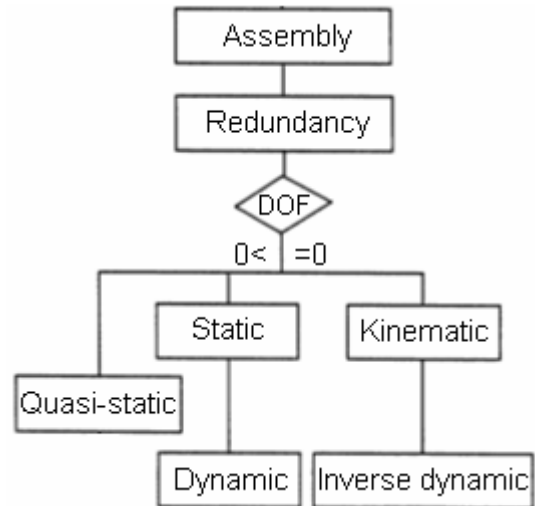


Fig. 2. Analysis flow-chart

The CAD software is used for creating the geometric (solid) model of the mechanical system. This model contains data about the mass & inertia properties of the rigid parts. The part geometry can be exported from CAD to MBS using standard format files, such as STEP or PARASOLID. To import the geometry of the rigid parts, the MBS software reads the CAD file and converts the geometry into a set of MBS geometric elements.

The FEA software is used for modeling flexible bodies in mechanical systems. Integrating flexibilities into model allows to capture inertial and compliance effects during simulations, to study deformations of the flexible components, and to predict loads with greater accuracy, therefore achieving more realistic results. The flexible body characteristics are defined in a finite element modeling output file (MNF - Modal Neutral File). The information in a MNF includes location of nodes and node connectivity, nodal mass and inertia, mode shapes, generalized mass and stiffness for modal shapes. The MBS model transmits to FEA the motion & load states in the mechanical system, which can be defined using a FEA Loads format file.

In the modern concept, the mechanical systems are approached as mechatronic systems, which integrate mechanics, electronics, and information technology. The mechatronic systems are built-up with some units with basic functions, which are made to interact between them in order to form a complex system with a given functionality. Integrating the control system in the mechanical model at the virtual prototype level, the mechanical designer and the controls designer can share the same model; they can also verify from one database the combined effects of a control system on a nonlinear, non-rigid model. In this way, the physical testing process is greatly simplified, and the risk of the control law being poorly matched to the physical (hardware) prototype is eliminated.

Therefore, for the mechatronic systems' design & simulation, the virtual prototyping platform has to include a DFC software solution, which directly exchanges information with the MBS software; the output from MBS is input for DFC and vice-versa. The mechanical model and the control system communicate by passing state variables back and forth. The analysis process creates a closed loop in which the control inputs from the control application affect the MBS simulation, and the MBS outputs affect the control input levels in DFC. In the concurrent engineering concept, the simulation of the mechatronic system involves the following steps: designing the mechanical model (including bodies, joints, forces etc.), analyzing the dynamic model, identifying the inputs and outputs that complete a closed loop between the MBS model and the control application, explaining the trajectory and synthesizing the reference signals, defining the input block diagram, designing the control system diagram, designing the controller and the interface electric circuits, and simulating the mechatronic system.

The analysis flow chart of the mechanical systems, shown in figure 2, depends on the degrees of freedom (DOF) of the system, which are given by the difference between the number of allowed motions and the number of geometric and kinematic constraints, according with the Gruebler count: $DOF = 6n - \Sigma(r_g+r_k)$, where n is the number of mobile bodies, r_g - the geometric restrictions, r_k - the kinematic restrictions.

Assembly analysis allows assembling all the parts in a system by joints. Input to the analysis is a set of measured positions and orientations of all parts from design draft of the mechanical system. Output from the analysis is a set of those values that minimizes constraint errors. Redundancy analysis is to eliminate redundant constraints from an over-constrained mechanism. Input to the redundancy analysis is the assembled configuration of the mechanism. Output is a remodeled system without redundant constraints.

Kinematic analysis is to calculate time history of motion without considering forces and mass effects on the motion. Inputs to kinematics are the assembled configuration of the mechanical system and the time dependent driving constraints. Outputs from the kinematics are time histories of positions, velocity and acceleration.

Static analysis is to find a stable equilibrium configuration with zero velocity and acceleration of the mechanical system. Inputs to analysis are positions, orientations of all parts with the forces acting on them. Output from the static analysis is the static equilibrium configuration (the positions and the orientations of the parts), and the reactions forces. Quasi-static analysis is a series of static equilibrium analyses for different loading conditions of forces or constraint values.

Dynamic analysis is to calculate time history of the motion due to forces acting on the mechanical system. Inputs to the dynamics are external and internal forces, and the assembled configuration. Outputs from the dynamics are time histories of positions, velocities, accelerations of the parts and the reaction forces. Inverse dynamic analysis is to determine the applied forces that are required to generate the prescribed motion of a kinematic system. Input to the analysis is a zero DOF system with mass properties and forces elements defined. Outputs from the inverse dynamic analysis are the same as in the dynamic analysis.

3. THE PHASES OF THE VIRTUAL PROTOTYPING PROCESS

As was mentioned, the main component of the virtual prototyping platform is the mechanical systems analysis and simulation software (MBS). The steps to create a virtual model with MBS software mirror the same steps to build a physical prototype:

- build – modeling bodies (parts), constrain the bodies, create forces - torques acting on bodies;

- test – measure characteristics, perform simulation, review animation, review numeric results as plots;
- validate – import experimental data, compare virtual and experimental results;
- refine – add friction, define flexible bodies, implement force functions, define controls;
- parameterize – add parametrics, define design variables;
- optimize – perform manual studies, design sensitivity studies, design of experiments, and optimization studies.

During the build phase, virtual prototypes are created of both the new product concept and any target products which may already exist in the market. The geometry and mass properties of the bodies are obtained from component solid models. The structural, thermal and vibratory characteristics result from component finite element models or experimental tests. One of the most important axioms for successful functional virtual prototyping is to simulate as you test. Testing of hardware prototypes has traditionally involved both lab tests and field tests in various configurations, which are very expensive. With virtual prototyping, it is enough to create virtual equivalents of the lab tests and the field tests, for example virtual test tracks in automotive simulation or virtual landing strips for aircrafts, and this cut time and cost.

To validate the virtual prototype, the physical and virtual models are tested identically (the same testing and instrumentation procedures are used both in the physical and virtual test process). The results are compared, and design sensitivity analyses are performed on the virtual model to identify design parameters that have great influence on the performance results that do not correspond. Afterwards, a lot of changes on the main design variables are realized in order to obtain an acceptable correlation between the virtual and hardware (physical) prototypes of the mechanical system.

Refining the virtual prototype involves the fidelity of the model. Replacing the rigid components with flexible counterparts, adding frictions, and representing the automatic systems that control the operating performance of the mechanical system can make the improvement of the virtual prototype.

The optimization of the virtual prototype is made with the following steps: parameterizing the model; defining the design variables; defining the objective function for optimization; performing design study and design of experiments; optimizing the model on the basis of the main design variables. Parameterizing the model simplifies changes to model because it helps to automatically size, relocate and orient bodies. In this way, relationships into the virtual model can be built, so that when a modeling object is changed, the MBS software updates any other objects that depend on it.

Design variables allow creating independent parameters and tie modeling objects to them. In addition, by using design variables, the parametric analyses can be performed: design study, design of experiments and optimization. Design optimization represents the capability to define design objectives, constraints and variables, and then the MBS software iterates automatically to the optimally - performing configuration. Design of experiments (DOE) is a complementary technique to design optimization. DOE is a methodology for running a statistically significant battery of tests on a design to determine its sensitivity or robustness to design or manufacturing variations. Design study describes the ability to select a design variable, sweep that variable through a range of values and then simulate the motion behavior of the various designs in order to understand the sensitivity of the overall system to these design variations.

In this approach, complex virtual prototypes for different mechanical & mechatronic systems (products) can be developed, which exactly replicate the structure (components) and the operating conditions of the product. The virtual prototyping technique has become very important in a lot of applications specific to mechanical engineering field, such as

automotive industry, robotics, biomechanics, or aircraft industry. In the field of automotive design & development, the virtual prototyping is used in different types of applications, as follows: suspension design (predicting suspension characteristics, optimizing suspension design, loads analysis, establishing wheel envelopes, packaging studies); vehicle dynamics (modeling tire-roadway interaction, simulating linear-range and emergency handling maneuvers, predicting vehicle stability, braking studies, predicting chassis behavior on acceleration, assessing durability); engine design (valve train dynamics, timing chain design and simulation, crankshaft loads prediction); power train engineering (transmissions, transfer cases, differentials, drivelines, linkage design, predicting shift linkage precision and driver effort, analyzing transmission gear rattle and shifting smoothness, predicting bearing loads); body hardware engineering (door, trunk, and hood latch design, trunk and hood hinge linkage design, windshield wiper simulation and refinement, seat mechanism design, sun-roof and convertible mechanism design, window mechanism design). In the next section of the paper, the design and simulation of an active suspension system is presented for demonstrating the modeling & analysis capabilities of the virtual prototyping technique.

4. CASE STUDY – ACTIVE SUSPENSION SYSTEM

A classic suspension system consists of an energy dissipating element, which is the damper, and an energy-storing element, which is the spring [10]. Since these two elements cannot add energy to the system this kind of suspension systems are called passive; the advantages are simplicity and costs. If there a force actuator is placed in parallel to passive system, an active suspension system is obtained. The active systems use sensors to measure the accelerations of sprung mass and unsprung mass, the analog signals from the sensors being transmitted to the controller; in this way, the passenger comfort and car stability can be improved [7, 11, 12].

The study in this section aims a tradeoff between passenger comfort, i.e. minimizing car body travel, versus suspension travel as the performance objective. The idea is to transform the conventional passive suspension into an active one, using a hydraulic actuator, controlled by feedback, between the chassis and wheel assembly. The suspension system is approached in mechatronic concept, by integrating the mechanical structure and the electronic control system at the virtual prototype level. In fact, the virtual prototype is a control loop composed by the multi-body mechanical model connected with the dynamic model of the actuator and with the controller model. The mechanical model of the suspension mechanism was developed by using the MBS environment ADAMS, while for the control system design we used the DFC software solution EASY5.

The study is developed for a quarter-car model, which contains the guiding & suspension system of the left front wheel. The MBS model of the passive suspension system is shown in figure 3. A four-bar mechanism is used for the suspension of the wheel. The suspension linkage uses two control arms to hold the wheel carrier and control its movements. The lower and upper wishbones connect to the car body mount part using compliant joints (i.e. bushings). Spherical joints constrain the upright parts to the upper and lower control arms. Tie rod attaches to the steering mount part and to the wheel carrier through spherical joints. Revolute joints connect the wheel carriers to the tire mount part. The upper and lower struts of the damper (which are used to model the damper mass) are connected through cylindrical joints, and to the adjacent parts (car body, upper control arm) through spherical joints. The spring is disposed between the lower strut of the damper and car body. For limiting the suspension stroke, the model includes non-stationary bumper and rebound elements, which are disposed inside the damper.

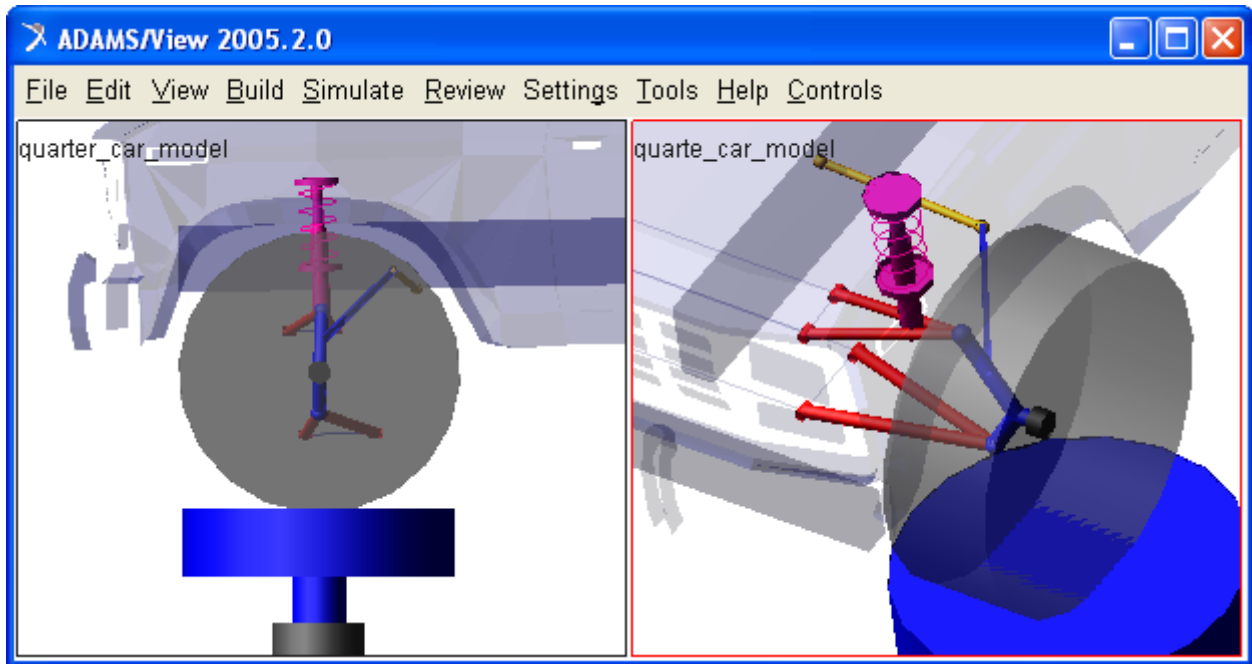


Fig. 3. The quarter-car suspension model

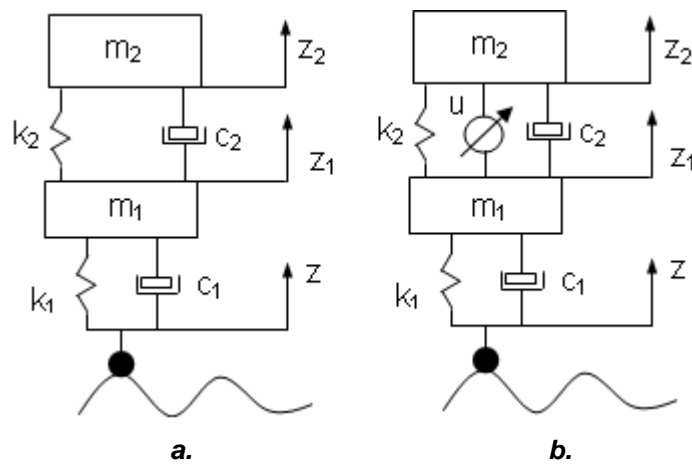


Fig. 4. The simplified quarter-car suspension model (a – passive, b – active)

As was mentioned, this is a quarter-car model, for which the car body equilibrium is assured with a translational joint to ground (i.e. fixed part), in the median plane of the vehicle, along the vertical axis. The simplified model of the quarter-car passive suspension is shown in figure 4.a, while the equivalent active suspension system is one in figure 4.b. The sprung mass m_2 represents the quarter car body, while the unsprung mass m_1 represents the wheel assembly (including the guiding mechanism). The spring k_2 and damper c_2 represent a passive spring and shock absorber that are placed between the car body and the wheel assembly, while the spring & damper group k_1 - c_1 serves to model the tire. The variables z_2 , z_1 , and z are the car body travel, the wheel travel, and the road disturbance, respectively. The force element, applied between the sprung and unsprung masses, is controlled by feedback and represents the active component of the suspension system.

According with the Newton-Euler formalism, the differential dynamic equations for the active suspension system can be written in the following form:

$$m_2\ddot{z}_2 = -c_2(\dot{z}_2 - \dot{z}_1) - k_2(z_2 - z_1) + u, \quad (1)$$

$$m_1\ddot{z}_1 = c_2(\dot{z}_2 - \dot{z}_1) + k_2(z_2 - z_1) - u + c_1(\dot{z} - \dot{z}_1) + k_1(z - z_1),$$

where “u” is the control signal (i.e. the force generated by the actuating system). The equations can be rewritten using the Laplace transformation,

$$(m_2s^2 + b_2s + k_2)Z_2(s) - (c_2s + k_2)Z_1(s) = U(s), \quad (2)$$

$$-(c_2s + k_2)Z_2(s) + (m_2s^2 + (c_1 + c_2)s + (k_1 + k_2))Z_1(s) = (c_1s + k_1)Z(s) - U(s)$$

The passive suspension system can be modeled with the same equations, considering $u=0$. The inputs in the dynamic model are the road disturbance (z) and the control signal (u), while the outputs are represented by the car body travel (z_2) and the wheel travel (z_1). The performance index of the system refers to the difference between output travels ($z_2 - z_1$). In these terms, we obtained the specific transfer functions G_z and G_u .

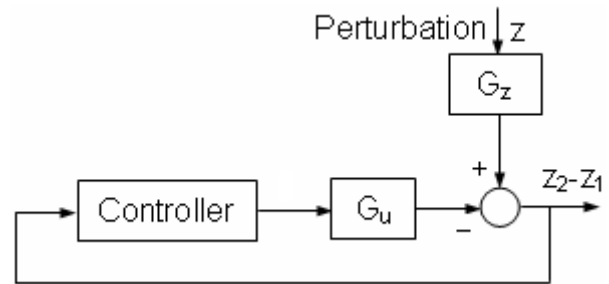


Fig. 5. The basic control scheme

The block diagram of the control system was designed considering the road disturbance as a perturbation, which must be eliminated, the basic scheme being shown in figure 5. The control system of the suspension system (fig. 6) was designed in concurrent engineering concept by using the DFC (Design for Control) software solution EASY5 of MSC Software. The input and output plants of the mechatronic system have been modeled in ADAMS/Controls, for connecting the mechanical model and the actuating - control system model. The communication data are saved in a specific file for EASY5 (*.inf); ADAMS/Controls also generates a command file (*.cmd) and a dataset file (*.adm) that are used during simulation [13]. In the control diagram, the “ADAMS Mechanism” block represents the MBS model of the suspension system (shown in figure 3), and it was created based on the information from the “inf” file.

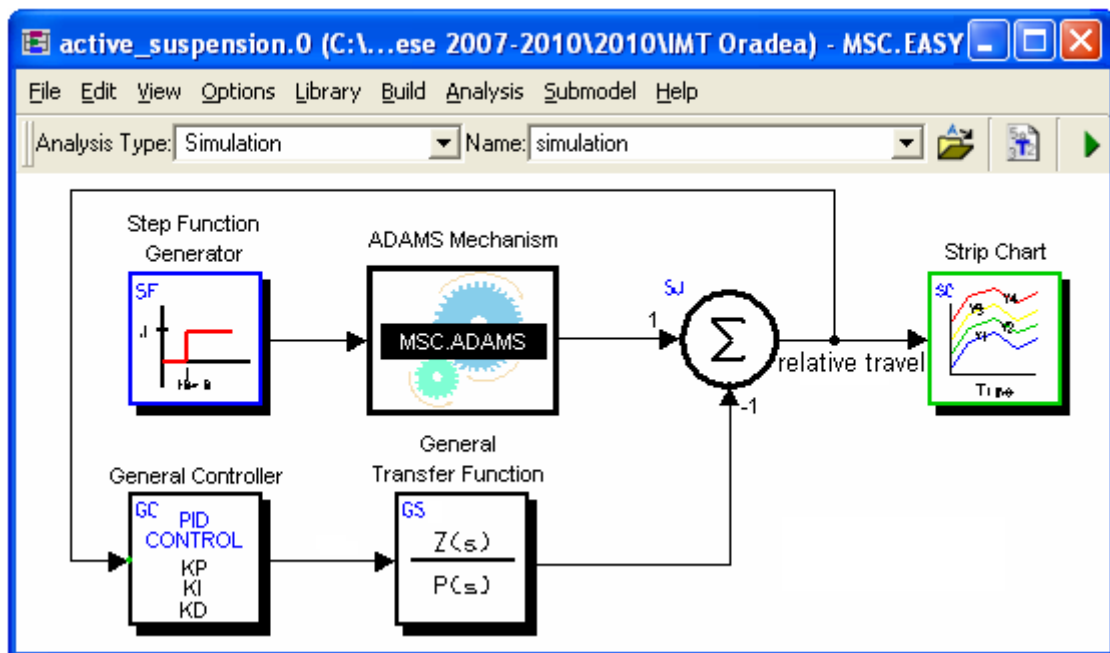


Fig. 6. The control system of the active suspension

From the controller point of view, for obtaining reduced transitory period and small errors, we used a PID controller, which attempts to correct the error between a measured process variable and a desired set-point by calculating and then outputting a corrective action that can adjust the process accordingly. The proportional gain determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The control law of the PID controller is given by the next equation:

$$v(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (3)$$

which can be rewritten using the Laplace transformation,

$$V(s) = \left[K_p + \frac{K_i}{s} + K_d s \right] E(s), \quad (4)$$

where K_p is the proportional term, K_i - the integral term, K_d - the derivative term, e - the position error, τ - the generalized force, V - the Laplace transformation for the command signal, E - the Laplace transformation for the position error. The output signal of the controller (v) will be sent to the plant, and the new output will be obtained. This new output will be sent back to the sensor again to find the new error signal (e); the controller takes this new error signal and computes its derivative and its integral again.

There are several methods for tuning a PID loop. The most effective methods generally involve the development of some form of process model, and then choosing P, I, and D based on the dynamic model parameters. To obtain the desired response, there is the following sequence: add a proportional control to improve the rise time, add a derivative control to improve the overshoot, add an integral control to eliminate the steady-state error. In our research, the tuning of the PID controller was made by using the EASY5 Matrix Algebra Tool (MAT). This is an interactive tool for doing numerical calculations, which consists of a high-level language for writing algorithms, a multi-window graphical user interface, and several libraries of numerical functions. MAT is specifically designed to be used in conjunction with EASY5 to perform such tasks as control system design, model data preparation, and post processing of simulation results.

The variables used in study are the proportional, derivative and integral terms of the PID controller, while the objective of the optimization is to improve the rise time, the overshoot, and the steady-state error. For performing the optimization, the MSC.EASY5 model (shown in figure 6) was exported as EMX file. In this way, MAT and EASY5 are used together to design the controller and to pass data back and forth between the two programs. We performed the control system design using MAT, and then evaluated the performance of the controller and active suspension system model in EASY5 and ADAMS.

In MAT, we used the “minimize_v” function to perform the optimization; the function has the following syntax [14]: $[x,f] = \text{minimize_v}(\text{funcname}, x0, H0, \text{tol}, \text{delx})$, where: funcname is the name of function used to setup minimization, $x0$ - initial guess for minimizer (real N-vector), $H0$ - initial guess for Hessian (real NxN matrix), tol - relative tolerance for x (real scalar), delx - relative step size for computing gradients by differencing (real scalar), x - minimizer of the function or last search position (real N-vector), f - value of the function at x (real scalar). The next step was to create the MAT minimizer function that performs the optimization we require (concerning to the settling time, the overshoot, and the steady-state error). The optimization is performed by calling the “minimize_v” function with the minimizer function as the first argument; MAT will repeatedly call the function as it performs the minimization procedure. The function will set the rise time, the overshoot, and the steady-state error appropriately and returns the error in the simulation, defined as the

sum of the squares of the differences between the simulation and desired values. As MATLAB proceeds through the minimization, we will see the calculation converge; finally, the values of K_p , K_i and K_d will result in a simulation that meets the design requirements. The detailed tuning algorithm of the PID controller will be presented in a future paper.

5. RESULTS AND CONCLUSIONS

The numeric simulations, whose results are shown in figure 7, have been performed considering a step input signal, with the amplitude of 0.1 m, like the wheel would climb over a road border. For a comfortable suspension system, the settling time should be less than 2 seconds, while the maximum acceptable value for the overshoot is 5% [8]. Referring to the passive suspension, the stabilization time is good, but the overshoot is too high ($\sigma \cong 31\%$). Therefore the control system aims to decrease the overshoot, without adversely affecting the stabilization time. Following the control strategy for the active suspension, the overshoot of the output signal is very small ($\sigma \cong 1.5\%$), while the stabilization time is around 1 second. Therefore, the results for the active suspension are in the recommended field for a comfortable suspension system, and this justifies the viability of the control strategy.

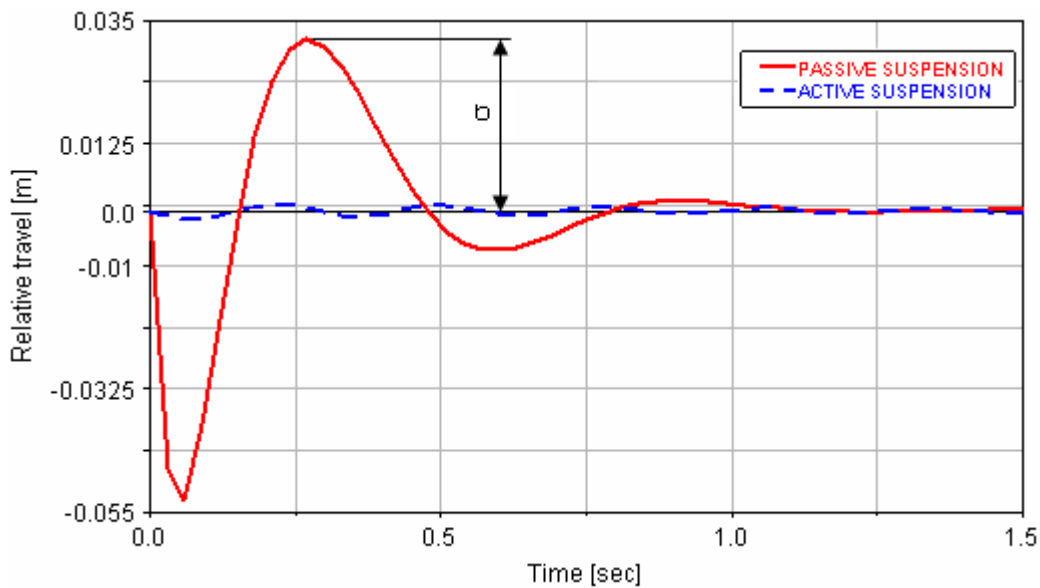


Fig. 7. Comparative results for the passive & active suspension systems

The application is a relevant example regarding the implementation of the virtual prototyping tools in the design process of the mechanic & mechatronic systems. One of the most important advantages of this kind of simulation is the possibility to perform virtual measurements in any point or area of the system, and for any parameter (motion, force, energy). Virtual prototyping-based optimization tools allow realizing the projected reductions in cycle times while maintaining and increasing the performance, safety, and reliability. This helps us to take quick decisions on any design changes without going through expensive hardware prototype building and testing. The behavioral performance predictions are obtained much earlier in the design cycle of the products, thereby allowing more effective and cost efficient design changes and reducing overall risk substantially.

Concerning the case study presented in paper, the future researches in the field will be focused on the verification of the controller behavior through frequency analysis methods, as well as the control system robustness (i.e. the capability of the control system

to operate with the imposed performance indexes, or closed-by these values, when one or more parameters of the physical model are changing). At the same time, we intend to extend the research for more complex suspension system models, such as half-car and full-car models, and for other control strategies & controller types (including the intelligent controllers, FLC - Fuzzy Logic Controllers).

ACKNOWLEDGMENTS

The paper was realized in the frame of the PN II - ID - PCE research grant, CNCSIS code 607/2009 (New structures for the suspension & steering systems of the motor vehicles, in mechatronic concept).

REFERENCES

- [1] Alexandru, C. Testing the Guiding - Suspension System of the Motor Vehicles in Virtual Environment. *International Review of Mechanical Engineering*, Vol. 3 (5), 2009, p. 521-528.
- [2] Bernard, A. Virtual Engineering: Methods and Tools. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 219 (5), 2005, p. 413-421.
- [3] Fischer, E. Standard Multi-Body System Software in the Vehicle Development Process. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-Body Dynamics*, Vol. 221 (1), 2007, p. 13-20.
- [4] Haug, E. J., Choi, K. K., Kuhl, J. G., Vargo, J. D. Virtual Prototyping Simulation for Design of Mechanical Systems. *Journal of Mechanical Design*, Vol. 117 (63), 1995, p. 63-70.
- [5] Hegazy, S., Rahnejat, H., Hussain, K. Multi-Body Dynamics in Full-Vehicle Handling Analysis. *Journal of Multibody Dynamics*, Vol. 23 (1), 1999, p. 19-31.
- [6] Höhne, G. et al.. *Extended Virtual Prototyping*. Springer, 2007.
- [7] Kruczek, A., Stribrsky, A. A Full-Car Model for Active Suspension - Some Practical Aspects. *Proceedings of the IEEE International Conference on Mechatronics - ICM*, Istanbul, 2004, p. 41-45.
- [8] Pozna, C. *Teoria Sistemelor Automate*. Ed. MATRIX-ROM, 2004.
- [9] Ryan, R. *Functional Virtual Prototyping - Realization of the Digital Car*. Mechanical Dynamics Inc. Press, Michigan, 2001.
- [10] Silva, M.M., Costa Neto, A. Handling Analysis of a Light Commercial Vehicle Considering the Frame Flexibility. *International Review of Mechanical Engineering*, Vol. 1 (4), 2007, p. 334-339.
- [11] Soliman, A. Active Suspension Systems for Road Vehicles - A Review. *Proceedings of the International Conference on Noise and Vibration Engineering - ISMA*, Leuven, 2004, p. 87-101.
- [12] Wang, J., Wilson, D., Xu, W., Crolla, D. Active Suspension Control to Improve Vehicle Ride and Steady-State Handling. *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, p. 1982-1987.
- [13] *** Getting Started using ADAMS. MSC Software Press, Santa Ana, 2005.
- [14] *** EASY5 Guide - Matrix Algebra Tool. MSC Software Press, Santa Ana, 2005.