SIMULATING THE DYNAMIC BEHAVIOR OF THE MECHANICAL SYSTEMS BY USING VIRTUAL PROTOTYPING TOOLS. CASE STUDY – SOLAR TRACKER SIMULATION

Cătălin ALEXANDRU

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

Abstract. In this paper, a lot of aspects regarding the dynamic analysis & simulation of the mechanical systems, by using virtual prototyping tools, are presented, as follows: the virtual prototyping phases, the components of the digital platform and the interactions between components. The approaching is made in the concurrent engineering concept, by integrating the control system in the mechanical model at the virtual prototype level (design for mechatronics). As case study, the virtual prototype of a solar panel tracking system is developed, using licensed software solutions: CATIA, ADAMS/View, ADAMS/Controls, and MATLAB/Simulink.

1. INTRODUCTION

Determining the real behavior of the mechanical systems is a priority in the design stage since the emergence of the computer graphic simulation. Important publications reveal a growing interest on analysis methods for multi-body systems that may facilitate the self-formulating algorithms, having as main goal the reducing of the processing time in order to make possible real-time simulation [3, 6]. These methods were used to develop powerful modelling and simulation environments - MBS programs, which allow building and simulating a computer model of any mechanical system that has moving parts. The mechanical device is treated as a constrained, multibody, spatial mechanical system, in which body elements are connected through mechanical joints and force elements. The main difference of mechanical system dynamics from the conventional structural system dynamics is the presence of a high degree of geometric nonlinearity associated with large rotational kinematics. Governing equations for conventional structural dynamics are linear differential equations, while those equations for mechanical system dynamics are nonlinear differential equations that are coupled with nonlinear algebraic equations of kinematic constraints.

As product complexity increases and competitive product development cycle times are reduced, hardware prototype creation and testing become major bottlenecks to successful new product launches. Due to these bottlenecks, leading global manufacturers are feeling increasing pressure to rapidly institute enterprise-wide, simulation-based design and virtual prototyping practices that can insure greater product performance and quality in a fraction of both the time and cost required with traditional build-and-test approaches. Virtual prototyping is a software-based engineering discipline that entails modeling a mechanical system, simulating and visualizing its 3D motion behavior under real-world operating conditions, and refining & optimizing the design through iterative design studies. This advanced simulation technique consists mainly in conceiving a detailed model and using it in a virtual experiment, in a similar way with the real case. An important advantage of this kind of analysis consists in the possibility of make virtual measurements in any point and area of the mechanical system and for any parameter.

Virtual prototyping allows the full operation of the product to be considered and evaluated early enough in the design process to allow for function to truly drive form and fit. It also allows multi-function optimization to be realized, such that a true balance can be obtained between competing functional requirements involving performance, safety,

durability, cost, comfort, etc. These two benefits were largely impractical in traditional development cycles involving extensive reliance on hardware prototypes. In addition to these benefits, functional virtual prototyping has proven effective in facilitating tighter and more successful relationships between manufacturers and their lead suppliers. Virtual Prototyping brings several advantages: it reduces the time and cost of the new product development; it reduces the product cycles; it reduces the number of expensive physical prototypes and experiment with more design alternatives.

Frequently used in mechanical systems engineering in the last decades, the classic CAD/CAM/CAE technologies were orientated on the concept "art-to-part" that is directed toward the design, development and manufacturing of the system's components [5]. Three-dimensional solid modelers were used to design the form of the components. Finite element programs performed detailed meshing and analysis of structural, thermal and vibratory characteristics of individual parts. At the same time, specialized software aimed at improving manufacturability of components. 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 mechanical system is a major contributor to overall product quality. The only way to increase quality, and reduce time and cost, consists now of virtual prototyping applied to system level.

The modern design process of the mechanical systems involves conceptual design, functional design, testing and optimization [7]. The conceptual design has as central objective to establish the best product concept (in the given conditions, by performing an efficient management of information picked by the science, technology, economy, market, culture, legislation, policy etc.), using specific design tools, such as multi-criteria analysis and morphologic analysis. The multi-criteria evaluation is for obtaining the most efficient and feasible product, performing the following steps: establishing the variants and the evaluation criteria; determining the contribution for each criterion; applying the contribution notes for each variant; establishing the optimal variants by using the consequences matrix. The morphologic analysis is made for choosing the solution by taking into consideration the requirements list, which contains aspects regarding the material, form, color, mounting place etc. For describing the solutions, there are used combinatory procedures, which generate the specific morphologic table.

The functional design involves identifying, modeling and evaluating the operational performances of the mechanical systems, and the deviations from the imposed characteristics, with other words the mode in which the mechanisms responses to the design requirements. The present-day researches in the field of kinematics and dynamics of mechanical systems are focusing upon several topics such as the development of new formalism for the description of the mechanical systems, the improvement of classical methods with new mathematical theories, the development of new analysis programs, and the improvement of the numerical integration methods [3].

2. THE VIRTUAL PROTOTYPING PLATFORM

Generally, a typical virtual prototyping platform [1] includes the following software solutions (fig. 1): CAD - Computer Aided Design (ex. CATIA, PROENGINEER, SOLIDWORKS); MBS - MultiBody Systems (ex. ADAMS, SD-EXACT, PLEXUS); FEA - Finite Element Analysis (ex. NASTRAN/PATRAN, NISA, COSMOS, ANSYS); Command & Control (ex. MATLAB, EASY5, MATRIX). In addition, a modern virtual prototyping platform includes specific software for product data management (PDM); this technique is the glue that enables the main components of the virtual prototyping platform to be successful by making all of the up-to-date component data readily available and manageable.
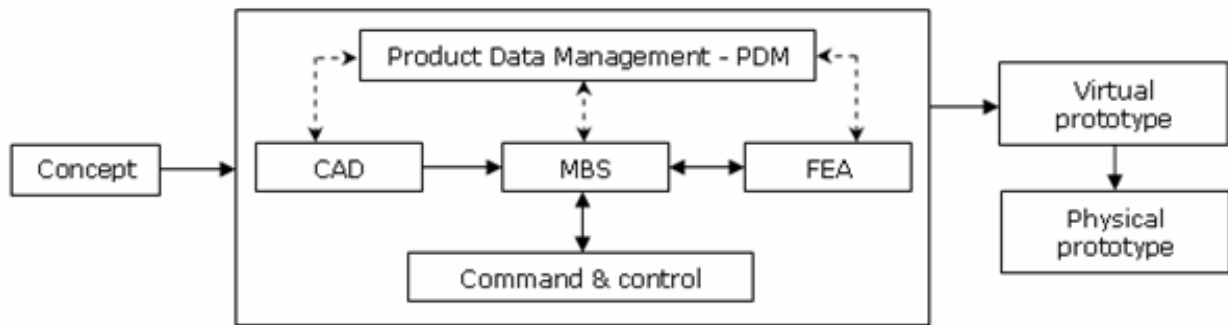
Fig. 1 The virtual prototyping platform

The MBS software is the main component of the virtual prototyping platform, and it allows analyzing, optimizing, and simulating the mechanical system under "real" operating conditions. The CAD software is used for creating the geometric model of the mechanical system (i.e. the solid model). This model contains information about the mass and the inertia properties of the rigid parts. At the same time, the CAD environment provides the ability to perform simple motion studies and to easily transfer geometry to the virtual prototyping software. The geometry can be exported from the CAD environment to the MBS environment using standard format file, for example IGES (Initial Graphics Exchange Standard) or STEP (Standad for the Exchange of Product Model Data). 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 flexibilities (flexible parts and compliant joints). Virtual prototyping provides the ability to transfer loads from MBS to FEA and to bring component flexibility from FEA back into MBS. Integrating the flexibilities into model allows capturing inertial and compliance effects during handling and comfort simulations, study deformations of the flexible components, and 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).

In the typical design process with controls, the mechanical designer and the controls designer work from the same concept, but use different sets of software tools. The result is that each designer produces a model for the same problem. Each design is then subject to verification and testing, and the first time the two designs are brought together is during physical prototype testing. If a problem occurs during the interaction between the controls design and the mechanical design, the engineers must refine the control design and/or the mechanical design, and then go through the entire verification process.
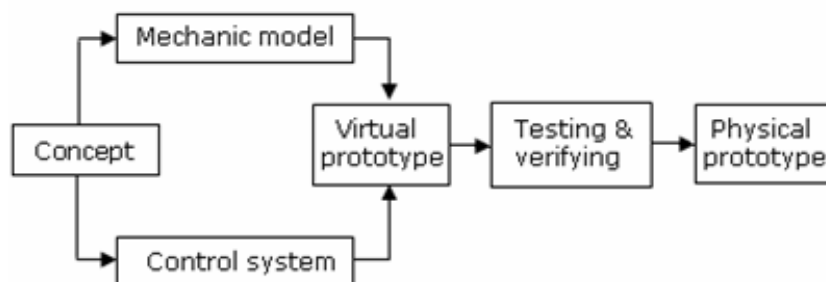

Fig. 2 The concurrent engineering model

Integrating the control system in the mechanical model (concurrent engineering), the two designers can share the same mechanical model (fig. 2); they can also verify from one database the combined effects of a control system on a nonlinear, non-rigid model [4]. The

physical testing process is greatly simplified, and the risk of the control law being poorly matched to the real system is eliminated. Inside the virtual prototyping platform, the command & control software directly exchanges information with the MBS software; the output from the MBS model is input for the control system and vice-versa.

In the concurrent engineering philosophy, the simulation algorithm of the mechanical systems involves the following steps: regarding the MBS software - designing the mechanical model (including bodies, joints, elastic and damping elements, external forces and torques); analyzing the dynamic model, identifying the inputs and outputs, which complete a closed loop between the MBS model and the control application, exporting the obtained model; regarding the control application - importing the mechanical model block, explaining the mechatronic system trajectory in the task space and defining the input block diagram, designing the control system block diagram, designing the controller and the interface electric circuits, simulating the mechatronic system. This simulation 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 [8].

The steps to create a virtual (software) model mirror the same steps to build a physical (hardware) model, as follow: building - modeling parts, constrain the parts, create forces; testing - measure characteristics, perform simulation, review animation; validate - import test data, superimpose test data; refine - add friction, define flexible parts, define control; optimize - add parametrics, define design variables, define objective functions, perform design studies, and perform 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.

To validate the virtual prototype, the physical and virtual models are tested identically, using the same testing and instrumentation procedures. 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 models.

Refining the virtual prototype involves the fidelity of the model. By replacing the rigid components with flexible counterparts, adding frictions, and representing the automatic systems that control the operating performance of the mechanical system, respectively, 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. Design variables allow creating independent parameters and tie modeling objects to them. 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. Design optimization represents the capability to obtain the optimal configuration of the system.

3. CASE STUDY

In the last years, the virtual prototyping has become very important in a lot of applications in the field of renewable energy systems design and development, such as solar tracking systems, wind turbines, and thin layer deposition stands [2]. The solar energy conversion is one of the most addressed topics in this field. The efficiency of the solar system depends on the degree of use & conversion of the solar radiation; when performing the energy balance, the reference is done to the surface that absorbs the incoming radiation and to the balance between energy inflow and energy outflow.

The rate of useful energy leaving the absorber is given by the difference between the rate of optical (short wavelength) radiation incident on absorber and the rate of energy loss from the absorber. In these terms, there are two ways for maximizing the rate of useful energy: optimizing the conversion to the absorber level, and decreasing the losses by properly choosing the absorber materials; increasing the incident radiation rate by using mechanical tracking systems.

Basically, the tracking systems are mechanical devices (i.e. mechanisms), driven by motors or actuators, which orient the panel in order to follow the sun path on the sky. The orientation of the solar panels may increase the efficiency of the conversion system up to 50%. The orientation principle is based on the input data referring to the position of the sun on the sky dome. For the highest conversion efficiency, the sunrays have to fall normal on the receiver so the system must periodically modify its position in order to maintain this relation between the sunrays and the panel. The positions of the Sun on its path along the year represent an input data in designing the solar trackers. Consequently, for the design process of the tracking systems, there are taken into consideration two rotational motions: the daily motion, and the yearly precession motion.

In these conditions, there are two fundamental ways to track the sun - by one axis, or by two axes. The single-axis tracking systems pivot on their axis to track the sun, facing east in the morning and west in the afternoon. The tilt angle of this axis equals the latitude angle of the loco because this axis has to be always parallel with the polar axis. In consequence for this type of tracking systems is necessary a seasonal tilt angle adjustment. The two-axis trackers combine two rotational motions, so that they are able to follow very precisely the sun path along the period of one year. That's why dual axis tracking systems are more efficient than the single one, but also more expensive because they are using electrical and mechanical parts. In this case, there are two independent motions: the daily motion, and the seasonal motion, respectively.

To demonstrate the virtual prototyping capabilities, in the present paper the virtual prototype of a dual-axis solar tracking system is developed. The solar tracker is designed for following the sun in both polar and seasonal directions, keeping the sun's rays normal to the platform surface. A rotary motor that is able to develop an angular displacement up to 180 degrees directly drives the daily motion. A hydraulic actuator that changes the panel elevation drives the seasonal motion.

The 3D-solid model was made using CAD software - CATIA. The geometry was transferred to ADAMS/View using the STEP file format, through ADAMS/Exchange module [8]. The virtual model of the tracking system, shown in figure 3, is designed so that it has five parts, which are connected one with other, respectively to ground, using the following geometric constraints: base (1) fixed on the ground; intermediary element (2) mounted on the base – revolute joint; hydraulic cylinder (4) to base – revolute joint; hydraulic piston (5) to intermediary element – revolute joint; cylinder to piston – translational joint; motor fixed on the intermediary element; solar panel (3) to motor (and on the intermediary element) – revolute joint.
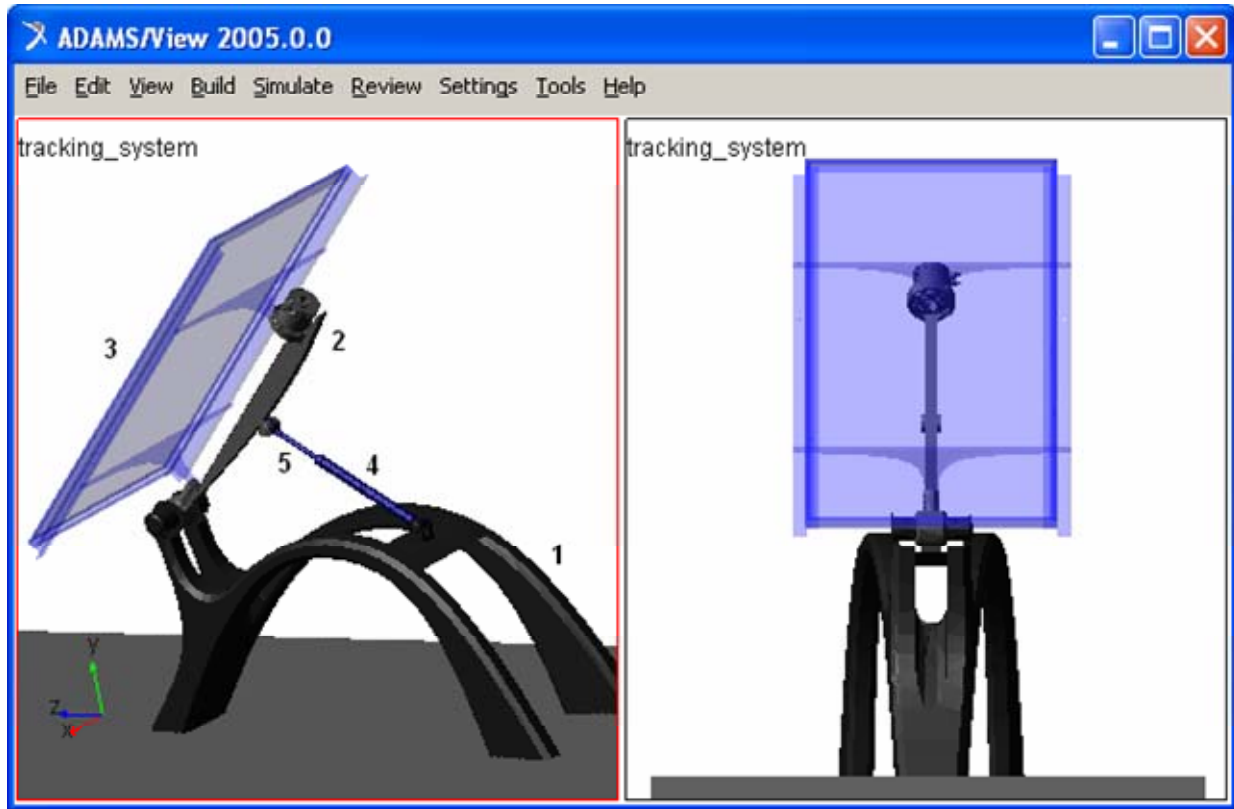
Fig. 3 The virtual prototype of the solar tracking system

For the kinematic model of the tracking system, there are two motion generators, which are applied in the revolute joint between intermediary element and the solar panel, and in the translational joint between the hydraulic piston and cylinder. For the dynamic model, the motion generators are replaced with external (motor) torque & force that drive the panel, which are computed by control systems, based on the error between the actual panel position and the desired position. This is more realistic than attaching motion generators and driving the motion directly; at the same time, we can look at issues related to motor size in an actual tracking system.

For this paper, only the control of the daily motion is considered, so that we are looking for the polar control torque that is applied to the solar panel. The control system is developed using ADAMS/Controls and MATLAB/Simulink. ADAMS/Controls is a plug-in to ADAMS/View that allows integrating motion simulation and controlling system design in the virtual model. ADAMS/Controls allow connecting the ADAMS mechanical model to block diagrams that are developed with control applications.

For driving the daily motion, a DC motor is used, so that the objective is to control the angular velocity of the rotor, which is perturbed with the motor torque. In fact, the angular velocity of the rotor determines the daily position (i.e. the polar angle) of the panel. The motor torque is obtained having in view the mechanical model of the tracking system, on which acts the external environment.

The four-step process of combining controls with a mechanical system involves: build the mechanical model, which includes all necessary geometry, constraints, forces; identify the ADAMS inputs and outputs - the outputs describe the variables that go to the controls application, and the inputs describe the variables that come back into ADAMS; build the control system block diagram, in MATLAB/Simulink, and include the ADAMS plant in the block diagram; simulate the combined mechanical model & control system.

In our case, the input to the solar tracking system model is the angular velocity of the rotor. The output, which will be transmitted to the controller, is the control torque. ADAMS/Controls and MATLAB communicate by passing state variables back and forth. Therefore, it is necessary to define the input and output variables of the model, and the functions that those inputs and outputs reference, with a set of ADAMS state variables. For the input state variable, representing the angular velocity of the rotor, the run-time function is 0.0 during each step of the simulation, because the velocity will get its value from the control application. The run-time function for the input variable is VARVAL(rotor_ velocity), where VARVAL is an ADAMS function that returns the value of the given variable. In other words, the input angular velocity gets its value from the input variable. For the output state variable, representing the control torque, the run-time function returns the sum of torques on body at location.

The next step is for exporting the ADAMS plant files for the control application. The Plant Input refers the input state variable (rotor_velocity), and the Plant Output refers the output state variable (control_torque). The input and output information are saved in a specific file for MATLAB (*.m). ADAMS also generates a command file (*.cmd) and a dataset file (*.adm) that will be used during the simulation. With these files, the control block was created in Simulink, in order to complete the link between the control and the mechanical system models (fig. 4).
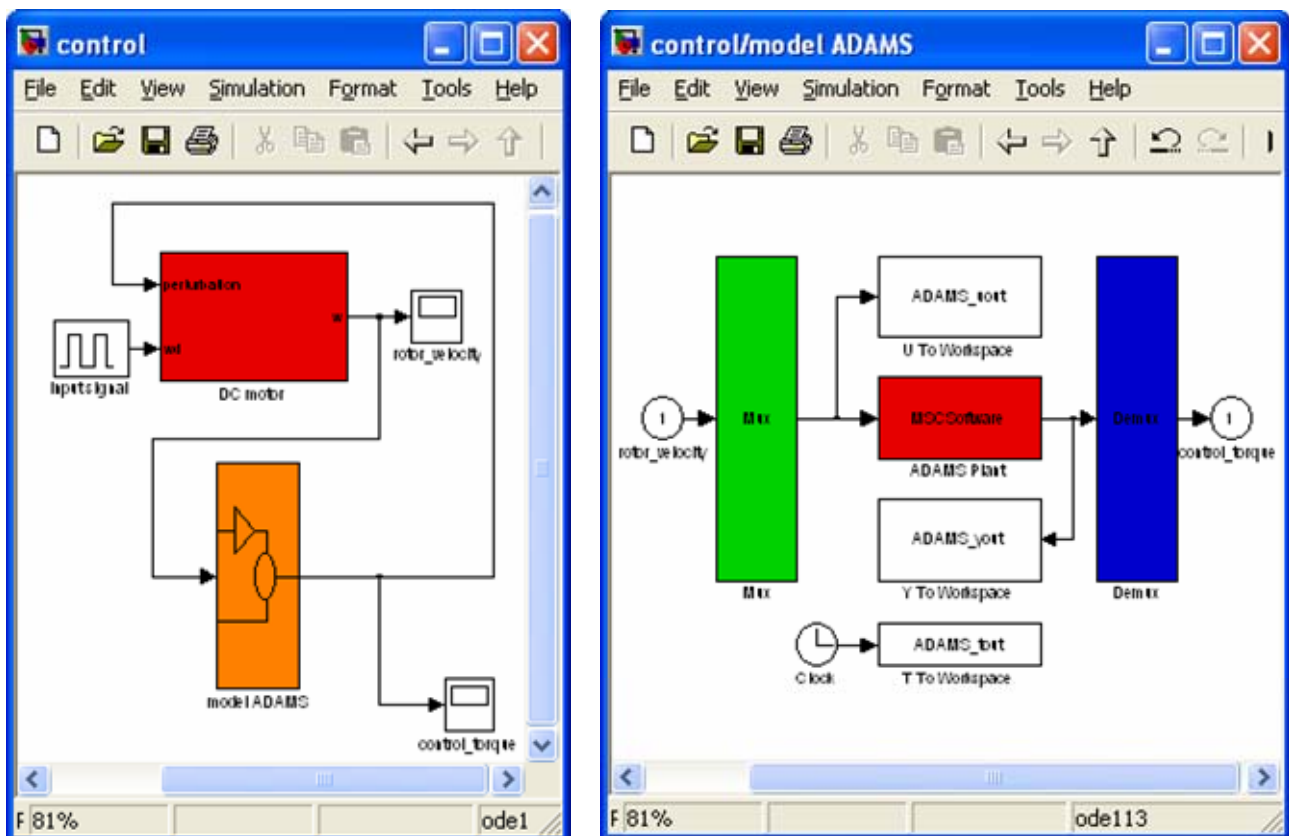

Fig. 4 The control system block (integrating ADAMS with MATLAB/Simulink)

The control system is designed in order to rotate the solar panel with equal steps (15 degrees at every hour), around the polar axis, and allows actuating brakes. These steps may have variable period of driving that allow an energy consumption comparison between different values of actuating time. ADAMS accepts the rotor velocity from MATLAB and integrates the mechanical model in response to them. At the same time,

ADAMS provides the control torque for MATLAB to integrate the Simulink model. Connecting the mechanical model with the control system, specific results are obtained in order to evaluate the dynamic behavior of the tracking system. For example, in figure 5, there are presented the time - history variations of the control torque and rotor velocity.
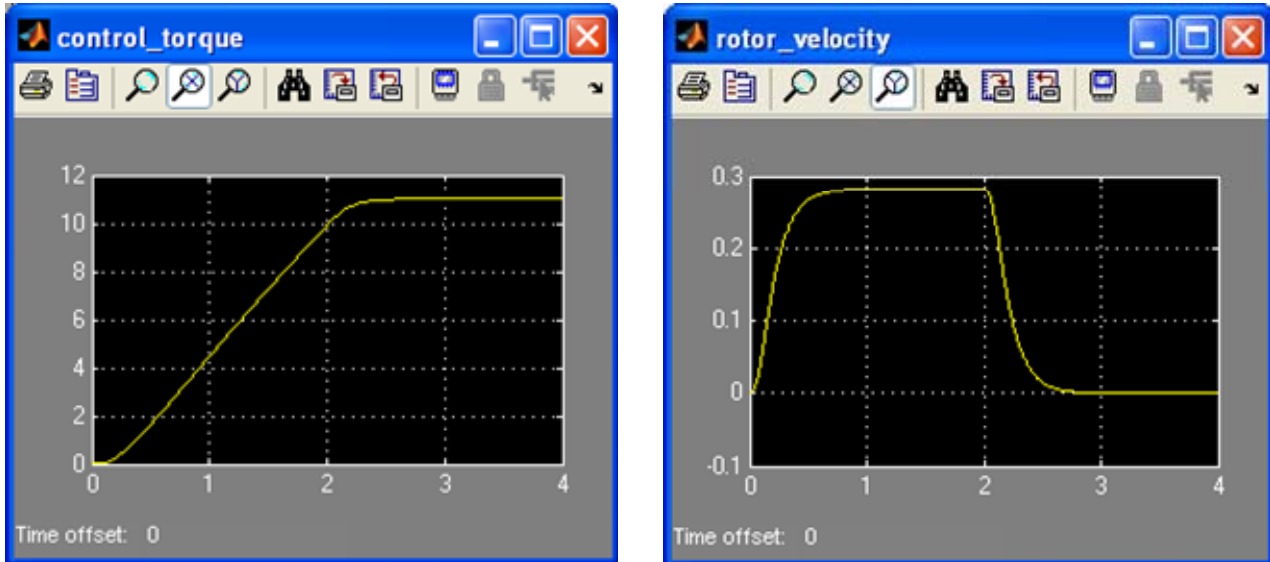

Fig. 5 Specific results for the dynamic analysis of the mechatronic system

5. CONCLUSIONS

Such results allow evaluating and optimizing the dynamic behavior of the mechanical system, making easy virtual measurements in any point and/or area, and for any parameter (displacement, velocity, acceleration, force and so on). This is not always possible in the real cases due to the lack of space for transducers placement, lack of appropriate transducers or high temperature. In this way, we can quickly exploring multiple design variations, testing and refining until optimizing mechanical system, without going through expensive physical prototype building and testing. Virtual prototyping-based design tools allow to more quickly assessing form, fit, function, and manufacturability of suspension – steering systems throughout concept design, concept refinement, detailed design, release, and production. Virtual prototyping allows realizing the projected reductions in cycle times while maintaining and increasing the vehicle performance, safety, and reliability.

REFERENCES
1. Alexandru, C., Pozna, C. Dinamica sistemelor mecanice pe baza prototipării virtuale, Editura Universităţii "Transilvania", Braşov, 2003.
2. Alexandru, C. Virtual prototyping of mechanical systems used to the renewable energy sources, Proceedings of the First Conference on Sustainable Energy – CSE, Braşov, 2005, p. 69 (Abstract Book).
3. Haug, E.J. Computer Aided Kinematics and Dynamics of Mechanical Systems, Allyn and Bacon, 1989.
4. Pozna, C. Teoria sistemelor automate, Editura MatrixRom, Bucureşti, 2003.
5. Ryan, R. Functional virtual prototyping, Mechanical Dynamics Publisher, Michigan, 2001.
6. Shabana, A. Dynamics of Multibody Systems, John Wiley and Sons, New York, 1988.
7. Vişa, I. Mechanical systems modelling as multibody systems in product design, Proceedings of the PRASIC'02 International Symposium, Brasov, vol. 4, p. 255-263, 2002.
8. *** Getting Started using ADAMS v2005, MSC Software, 2005.