

# Dynamic Modeling and Control of the Windshield Wiper Mechanisms

CATALIN ALEXANDRU, CLAUDIU POZNA

Product Design and Robotics Department

University Transilvania of Brasov

29 Bd. Eroilor, 500036 Brasov

ROMANIA

calex@unitbv.ro, cp@unitbv.ro

**Abstract:** - Determining the real behavior of the windshield wiper mechanisms is a priority in the design stage since the emergence of the computer graphic simulation. The paper attempts to analyze and simulate the windshield wiper systems by using a complex virtual prototyping platform, which integrates specific software solutions for realizing the 3D solid model (CATIA), modeling the MBS mechanical model (ADAMS), and controlling the mechanism (MATLAB). For applying the design strategy, we considered a tandem pattern double-lever system, which is approached in the concurrent engineering concept, by integrating the control system in the mechanical model at the virtual prototype level. To drive the mechanism, a DC motor is used, the objective being to control the angular velocity of the rotor, which is the input parameter in the mechanical model; the output, which is transmitted to the controller, is the motor torque. Driving the system by using the motor torque is more realistic than attaching a motion generator and driving the motion directly.

**Key-Words:** - Windshield wiper mechanism, Multibody system, Mechatronic model, Virtual prototype

## 1 Introduction

The windshield wiper mechanisms are vehicle-specific systems in which the wiping motion is transferred from the wiper motor to the pivot-shaft assemblies via linkages. A compact wiper system consists of the following components: wiper motor with thermo-switch, wiper gearing, motor crank, steel base-plate, crank linkage, pivot-shaft assembly with oscillating crank, and second pivot-shaft assembly with plate (for parallel wipe pattern). The linkage forces are supported by the sheet metal of the vehicle's bodywork.

For the modern vehicles, the following systems are frequently used (fig. 1): single-lever systems with parallel wipe patterns (a), single-lever systems with sector wipe patterns (b), opposed-pattern double-lever systems with parallel wipe patterns (c), opposed-pattern double-lever systems with overlapping sector wipe patterns (d), tandem-pattern double-lever systems with overlapping sector wipe patterns (e), and tandem-pattern three-lever systems with extra-wide overlapping sector wipe patterns (f).

The modern design process of the products (in this case, the wiper mechanisms) involves conceptual and functional design, command & control, digital mock-up, virtual prototyping and testing [1]. The conceptual design has as main objective to establish the best product concept, by using data picked by the science, technology,

economy or market, while the functional design involves identifying, modeling and evaluating the operational performances of the wiper systems, and the deviations from the imposed characteristics, with other words the mode in which the mechanism responses to the design requirements.

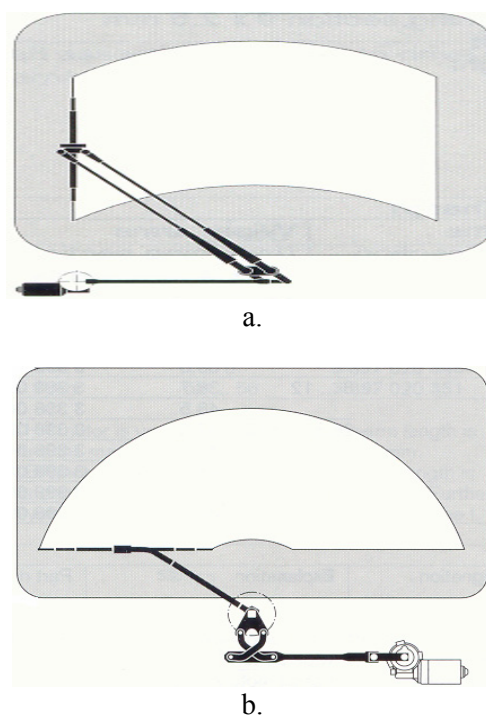
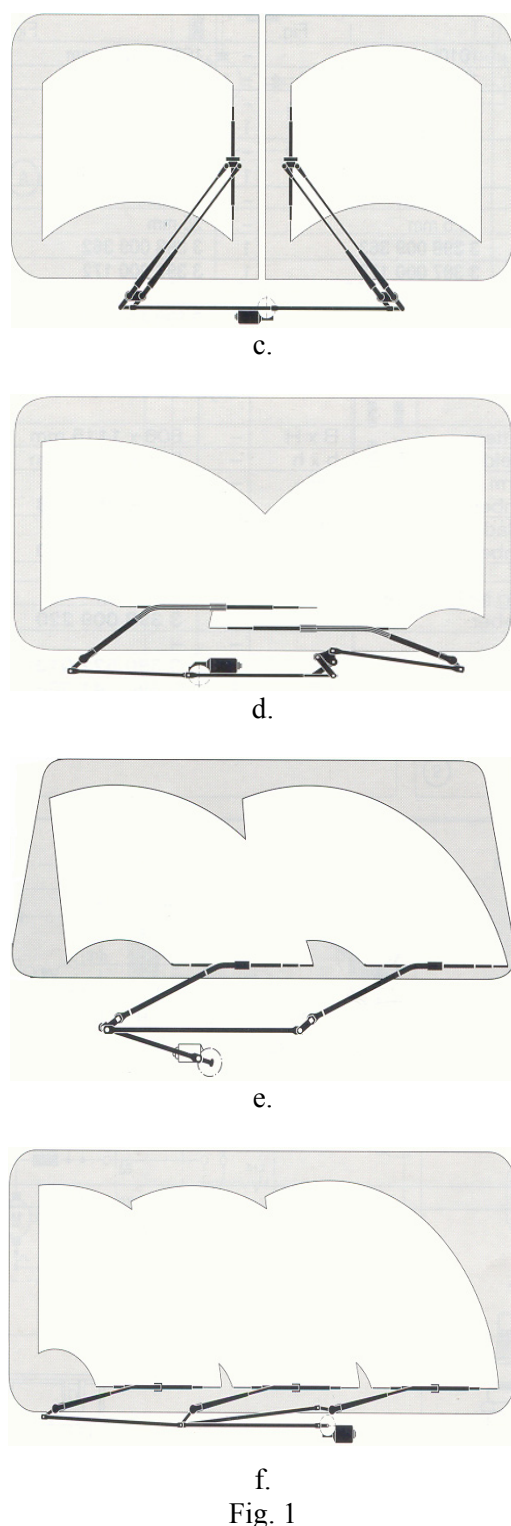


Fig. 1



In these terms, the paper attempts to analyze and simulate the windshield wiper mechanisms by using a virtual prototyping platform, which includes the following software solutions: CAD (Computer Aided Design), MBS (MultiBody Systems), FEA (Finite Element Analysis), and C&C (Command & Control). The application is made for a tandem pattern double-lever wiper mechanism, at which the driving element is a DC electric motor.

## 2 Design Steps and Tools

The analysis flow-chart of the windshield wiper mechanisms is shown in figure 2. The conceptual design has as objective to identify accurate and efficient mechanical configurations suitable for wiper systems. The synthesis can be made using a structural synthesis method based on Multi-Body Systems (MBS) theory, according to which a mechanical system is defined as a collection of bodies with large translational and rotational motions, linked by simple or composite joints.

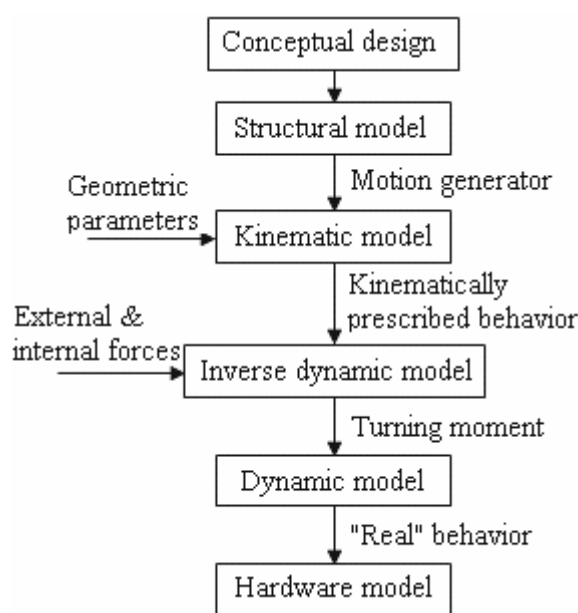


Fig. 2

The structural design consists in the following stages: identifying all possible graphs, taking into account the space motion of the system, the type of joints, the number of bodies, and the degree of mobility of the multibody system; selecting the graphs that are admitting supplementary conditions imposed by the specific utilization field; transforming the selected graphs into mechanisms by mentioning the fixed body and the function of the other bodies, identifying the distinct graphs versions based on the preceding particularizations, transforming these graphs versions into mechanisms by mentioning the types of geometric constraints (ex. revolute joint - R, or translational joint - T). The graphs of the multibody system are defined as features based on the modules, considering the number of bodies and the relationships between them (ex. R, T, R-R, R-T).

In the structural synthesis, there can be taken considered general criteria, for example the degree of mobility of the mechanism ( $M=1$ ), the number of

bodies, and the motion space ( $S=3$  for the planar mechanisms, and  $S=6$  in the general spatial case), as well as specific criteria, for example the type of the joints between the bodies. In this way, the structural synthesis method was applied and a collection of possible structural schemes were obtained, one of them being used in this study.

The kinematic model contains the rigid parts, which are connected through geometric constraints (joints), and the geometric parameters that define the mechanism; the input is made using kinematic restrictions (motion generators), which controls the angular position or velocity of the driving elements. The inverse dynamic model includes the kinematic model and, in addition, the external & internal forces; this model is used to determine the turning moment applied by the driving motor (in our case, the DC motor). The dynamic model includes the inverse dynamic model, but the input is made through the above-determined torque; the aim is to simulate/evaluate the real behavior of the windshield wiper mechanism.

Determining the real behavior is a priority in the design stage since the emergence of the computer graphic simulation. Recent publications [2, 3, 4] reveal a growing interest on analysis methods for multi-body systems, which may facilitate the self-formulating algorithms, having as main goal the reducing of the processing time in order to make possible real - time simulation.

The modeling & analysis of the windshield wiper mechanisms by using the MBS method involves the following steps [5]: defining the wiper mechanism as multi-body system (modeling the kinematic elements as bodies or geometric restrictions); establishing the coordinate frames of the bodies (global frame - for the ground, and local frames - one for each mobile part, respectively); defining the geometric model of the wiper mechanism, for each part in the local coordinate system; defining the connection matrices between the local coordinate frames and the global coordinate frame; establishing the global coordinates of the characteristic points that define the topologic model of the wiper mechanism; defining the equation system that describe the geometric and kinematic constraints from the wiper mechanism; formulating the motion's differential equations using different formalisms (ex. Newton - Euler, Lagrange); defining the mass and inertia characteristics of the bodies; establishing the reaction forces and torques from the wiper mechanism (elastic and damping forces / torques, joints reactions); solving the algebraic and differential equation system that describes the dynamic behavior.

In the last decade, a new type of studies was defined through the utilization of the MBS environments: the Virtual Prototyping [6, 7, 8]. This technology consists mainly in conceiving a detailed model and using it in a virtual experiment, in a similar way with the real case. No longer is it necessary to wait months to build a hardware prototype, instrument it, run tests on it, and make a small number of expensive modifications to it in order to assess proposed design changes.

The traditional CAD/CAM/CAE practices focused on a concept referred to as art-to-part. Nearly all software activities were oriented toward the design, development, and manufacturing of higher quality parts. Unfortunately, optimal part design rarely leads to optimal system design. The interaction of form, fit, function, and assembly of all parts in a product is a major contributor to overall product quality. The big opportunity to increase quality and reduce time and cost has now shifted to the system level. The system-level solutions include digital mock-up tools to investigate product form and fit, and virtual prototyping tools to assess product function and operating performance.

Basically, the virtual prototyping platform includes CAD, MBS and FEA programs. The CAD environment is used to create the geometric model of the system, which contains information about the mass & inertia properties of the rigid parts. The MBS software, which represents the central component of the virtual prototyping platform, is used for analyzing, optimizing and simulating the kinematic and dynamic behavior of the mechanical system. The FEA software is used for modeling flexible components, which allows capturing inertial and compliance effects during simulation, and predict loads with greater accuracy, therefore achieving more realistic results.

In the paper concept, the windshield wiper mechanisms 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. The component that increases the performance of the mechatronic system is the control software, which is integrated in the electronic part.

In the typical design process of a mechanical system 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. Integrating the control system in the mechanical model, the two designers 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 [9]. In this way, the physical testing process is greatly simplified, and the risk of the control law being poorly matched to the real (physical) system is eliminated.

In these terms, the virtual prototyping platform has to include a C&C (Command & Control) software product, which directly exchanges information with the MBS software; the output from the MBS model is input for the control system and vice-versa. 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 the concurrent engineering philosophy, the simulation algorithm of the windshield wiper mechanisms involves the following steps: developing the mechanical model (including bodies, joints, forces etc.); identifying the inputs and outputs that complete a closed loop between the mechanic model and the control system; importing the mechanical model block in the C&C environment; explaining the trajectory and synthesizing the reference signals; designing the control system diagram; designing the controller and the interface electric circuits; simulating the mechatronic system.

Under these circumstances, the general scheme of the virtual prototyping platform used in our research study is shown in figure 3. The digital platform integrates the following licensed software products: CAD - CATIA, MBS - ADAMS/View, FEA - ADAMS/AutoFlex, C&C - ADAMS/Controls & MATLAB/Simulink.

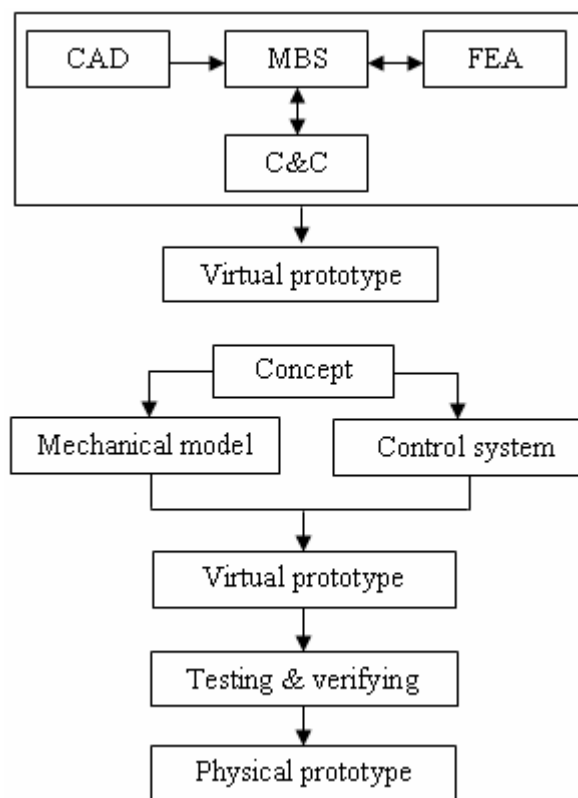


Fig. 3

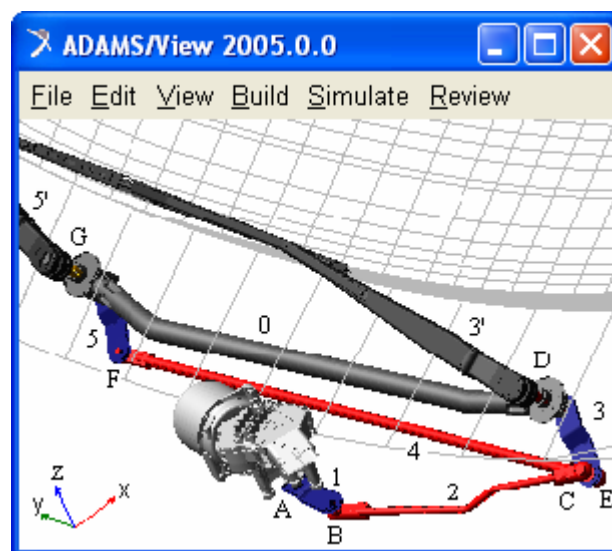


Fig. 4

### 3 Application

In order to apply the above-described analysis algorithm, a tandem pattern double-lever wiper mechanism (corresponding to a passenger car) was taken into consideration. The mechanism contains two four-bar spatial linkages: ABCD - the command mechanism, from the motor crank to the left wiper arm & lamella, and DEFG - the connection mechanism, which transmits the revolute motion to the right arm & lamella (fig. 4).

The 3D-solid model of the mechanism was made using the CAD software CATIA. The geometry was transferred to ADAMS/View using the STEP (Standard for the Exchange of Product Model Data) file format, through ADAMS/Exchange module. The elastic wiper lamellas were modeled using the specific ADAMS/AutoFlex module, by transforming the rigid geometry in flexible part.

The motor crank (1) and the left & right wiper arms (3, 5) are connected to the fixed part (i.e. the

car body - 0) using revolute joints (A, D, G). The crank linkages (2, 4) are connected to the motor crank, respectively to the wiper arms, using spherical joints (B, C, E, F). The passive rotations of the crank linkages are restricted by imposing null rotational motions in the joint C and E (kinematic restrictions). The mechanism has one degree of mobility,  $DOM=6 \cdot n - \sum r = 30 - 29 = 1$ , where  $n$  represents the number of mobile parts, and  $\sum r$  is the number of geometric & kinematic restrictions.

In the kinematic and inverse dynamic models, the mechanism is driven using a motion generator, which controls the angular position or velocity of the motor crank. For the dynamic model, the kinematic constraint is replaced by the motor torque applied to the input crank. Driving the mechanism by using the motor torque is more realistic than attaching a motion generator to the input crank and driving the motion directly. By applying a torque, we can look at issues related to motor size in an actual mechanical system.

The dynamic model includes also the friction forces that act on the flexible lamellas, which depend on the friction coefficient between rubber and windshield, and the normal force generated by the spring mounted between the wiper arm and the oscillating crank. The direction of the friction force depends on the sign of wiper arm velocity, which is modeled by the SIGN function using the specific Function Builder module. This function transfers the sign of an expression representing a numerical value to the magnitude of another expression, as follows:  $SIGN(a1, a2) = ABS(a1)$  if  $a2 \geq 0$ , and  $SIGN(a1, a2) = -ABS(a1)$  if  $a2 < 0$ , respectively. In our case,  $a1$  represents the magnitude of the friction force, and  $a2$  is the angular velocity of the wiper arm. In these circumstances, there are the following expressions for the left & right friction forces:  $SIGN(value, left\_arm\_velocity)$ ,  $SIGN(value, right\_arm\_velocity)$ , where  $value$  is the design variable that define the force magnitude.

In order to obtain the torque that move the motor crank, a 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 allows connecting the ADAMS model to block diagrams that are developed with the control application. The four-step process of combining ADAMS with MATLAB involves [10]: build the mechanical model, which includes all necessary geometry, constraints, forces; identify the plant inputs and outputs - the outputs describe the variables that go to MATLAB, and the inputs

describe the variables that come back into ADAMS; build the block diagram - build the control system block diagram with a control software, and include the ADAMS plant in the block diagram; simulate the combined mechanical model and control system.

To drive the windshield wiper mechanism, a DC electric motor is used, so that the objective of the controlling process is to control the angular velocity of the motor crank, which is perturbed with the crank turning moment. In these conditions, the input to the mechanical model is the angular velocity of the motor crank, and the output, which will be transmitted to the controller, is the crank turning moment - the motor torque (fig. 5).

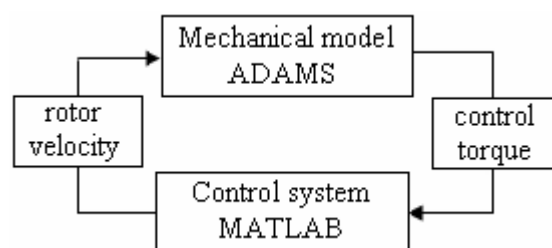


Fig. 5

The mechanical model and the control system communicate by passing state variables back and forth. Therefore, it is necessary to define the model's input and output variables, 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 motor crank, 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  $VARVAL(angular\_velocity)$ , where  $VARVAL$  (the acronym from Variable Value) is an ADAMS function that returns the value of the given variable [9]. In other words, the input angular velocity gets its value from the input variable. For the output state variable, representing the crank turning moment, the run-time function returns the sum of torques on motor crank in a coordinate system marker placed in the joint at car body. Thus, the function assigns the control torque to the output state variable.

The next step is for exporting the plant files for the control application. The Plant Input refers the input state variable ( $angular\_velocity$ ), and the Plant Output refers the output state variable ( $control\_torque$ ). ADAMS/Controls save the input and output information in an \*.m file (specific for MATLAB). It also generates a command file (\*.cmd) and a dataset file (\*.adm), which will be used during the simulation process.



With these files, the control system block is created in MATLAB/Simulink, in order to complete the link between the controls and mechanical systems. ADAMS/Controls and MATLAB communicate by passing state variables back and forth. ADAMS accepts the control inputs from MATLAB and integrates the mechanical model in response to them. At the same time, ADAMS provides the control torque information for MATLAB to integrate the Simulink model.

The mechatronic system is composed from the following components: the actuator, which in this case is a controlled DC motor, the gear box, the windshield wiper mechanism, the measurement system (i.e. the sensor), and the controller. The dynamic model of the DC motor (fig. 6), which is an electro-mechanical system, is given by the relations:

$$V = R_a i + L \frac{di}{dt} + E, J\epsilon_m = K_M i - b\omega_m - T, E = K_b \omega_m, \quad (1)$$

where  $V$  is the input voltage,  $T$  - the torque,  $R_a$  - the resistance,  $L$  - the inductance,  $i$  - the current,  $K_b$  - the contra-electromotor constant,  $q_m$  - the rotor position,  $\omega_m$  ( $\dot{q}_m$ ) - the angular velocity of the rotor,  $\epsilon_m$  - the angular acceleration,  $J$  - the inertial moment,  $K_M$  - the motor torque coefficient,  $b$  - the viscous friction [11].

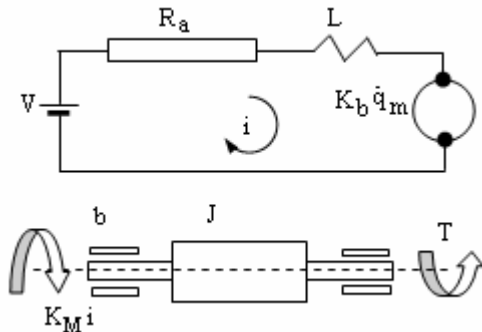


Fig. 6

The structure of the dynamic model is shown in figure 7, in which there are mentioned the interactions parameters between the three subsystems: DC motor with gear box, wiper mechanism, and the interaction with outside. The controlled parameter of the DC motor is the angular velocity, and for this reason the diagram from figure 7 is transformed in the diagram shown in figure 8.

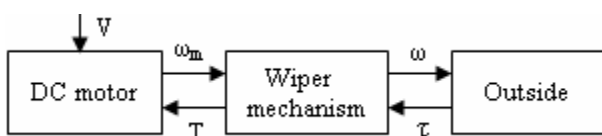


Fig. 7

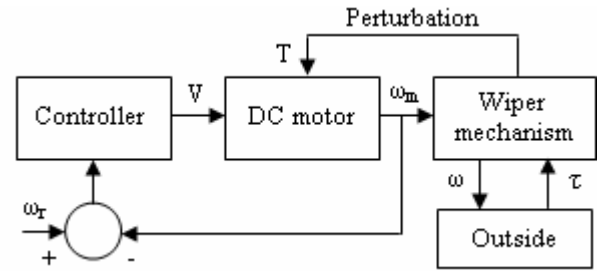


Fig. 8

According with this diagram, we will control only the DC motor, which is perturbed with the torque  $T$ . The computation of the torque  $T$  is based on the dynamic model of the mechanism on which acts the outside forces. For detailing the diagram shown in figure 8, we will focus on the DC motor subsystem, transforming the equations (1) with the Laplace operator, as follows:

$$\Omega_m(s) = G_V(s) \cdot V(s) - G_T(s) \cdot T(s), \quad (2)$$

where  $\Omega_m(s)$  is the Laplace transform of  $\omega_m$ ,  $V(s)$  - the Laplace transform of  $V$ ,  $T(s)$  - the Laplace transform of  $T$ ,  $G_V(s)$  &  $G_T(s)$  - the transfer functions:

$$G_V(s) = \frac{K_M}{LJs^2 + (JRa + bL)s + (bRa + K_b K_M)}, \quad (3)$$

$$G_T(s) = \frac{Ls + Ra}{LJs^2 + (JRa + bL)s + (bRa + K_b K_M)}. \quad (4)$$

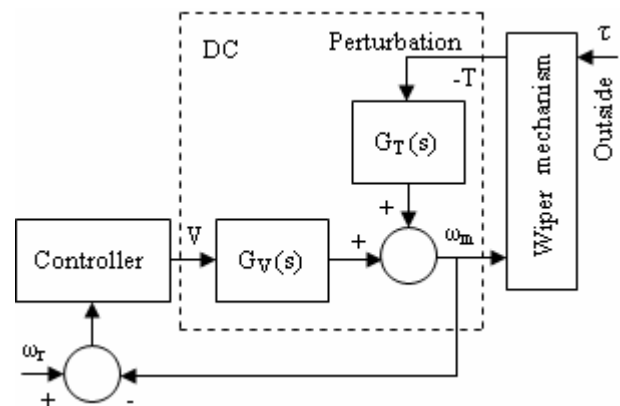


Fig. 9

In this way, we obtained the detailed model of the system, which is shown in figure 9. The transfer function of the plant is  $G_V$ , which has not poles in the origin, and for this reason the steady state error for a step input will be different from zero if we use a proportional controller. For obtaining a zero error steady state, it is necessary to use a proportional integer controller. By the final value theorem, we obtained the next result (see figure 10):

$$\varepsilon(\infty) = \lim_{s \rightarrow 0} s \cdot \frac{1}{s} \cdot \left( 1 - \frac{G_C G_V}{1 + G_V G_C} \right) = \lim_{s \rightarrow 0} \left( \frac{1}{1 + G_V G_C} \right) = 0. \quad (5)$$

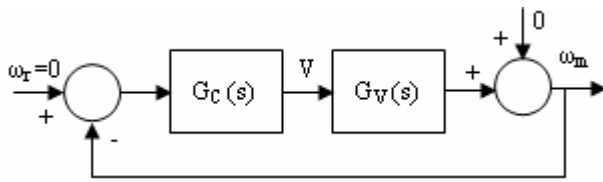


Fig. 10

If the perturbation is constant (approximated by the step signal) the steady state error can be computed with the relation (see figure 11):

$$\omega_m(\infty) = \lim_{s \rightarrow 0} \frac{1}{s} \cdot \frac{G_T}{1 + G_V G_C} = 0. \quad (6)$$

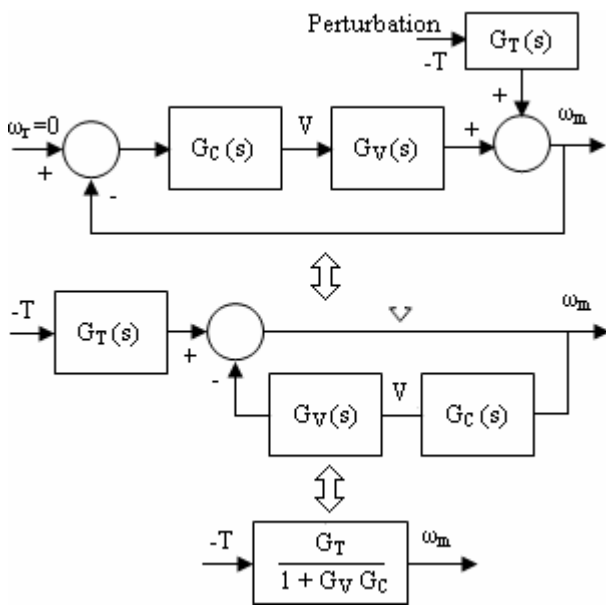


Fig. 11

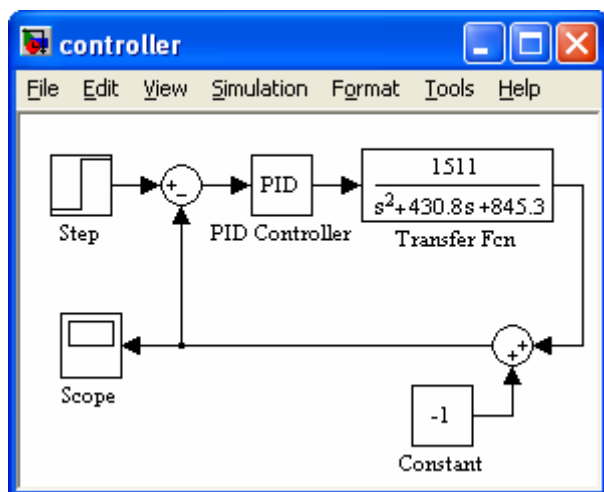


Fig. 12

After this comments, we have simulated the effects of a PI controller for the DC motor. The block diagram, realized in MATLAB/Simulink, is presented in figure 12. A more appropriated result can be obtained using the state space representation of the DC motor. The dynamical model of this system is given by the following equations, in which "x" represents the state vector:

$$\dot{x} = \begin{bmatrix} -\frac{b}{J} & \frac{K_M}{J} \\ -\frac{K_b}{L} & -\frac{R_a}{L} \end{bmatrix} \cdot x + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} \cdot u + \begin{bmatrix} -\frac{1}{J} \\ 0 \end{bmatrix} \cdot T, \quad (7)$$

$$y = [1 \ 0] \cdot x, \quad x = \begin{bmatrix} \omega \\ i \end{bmatrix}$$

In order to use the Ackermann algorithm, we have selected the poles position in:

$$\begin{cases} \gamma_1 = -15 \\ \gamma_2 = -14 - 0.2j \\ \gamma_3 = -14 + 0.2j \end{cases} \quad (8)$$

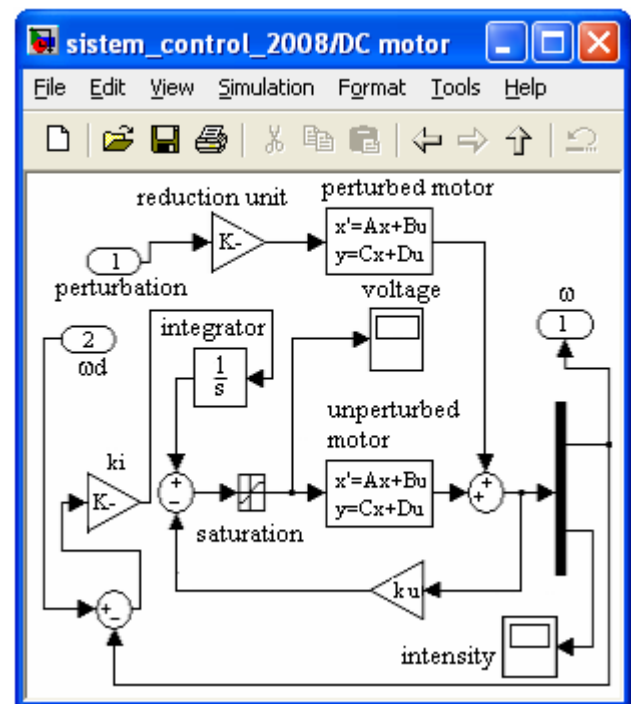


Fig. 13

In these terms, the simulation diagram was realized in MATLAB/Simulink (fig. 13), the simulation results being presented in figure 14. For the step response (fig. 14, a), the input is a step signal - the desired angular velocity is constant; the perturbation is zero, and the output is over damped signal, which has a zero steady state. For the

perturbation rejection (fig. 14, b), the input signal is zero and the perturbation is constant; in the end, the output is zero, and the perturbation is rejected.

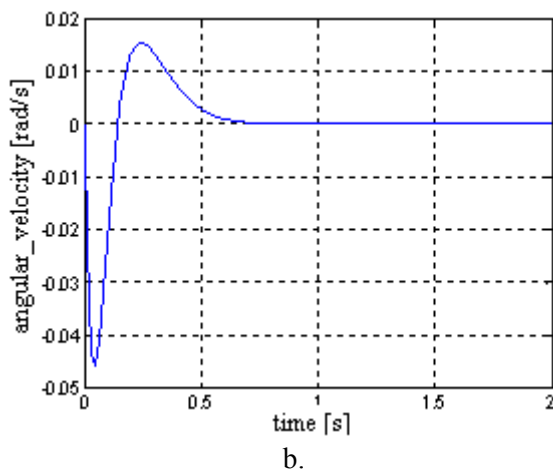
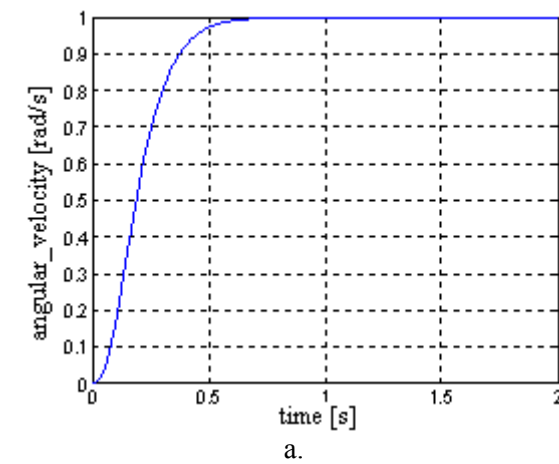


Fig. 14

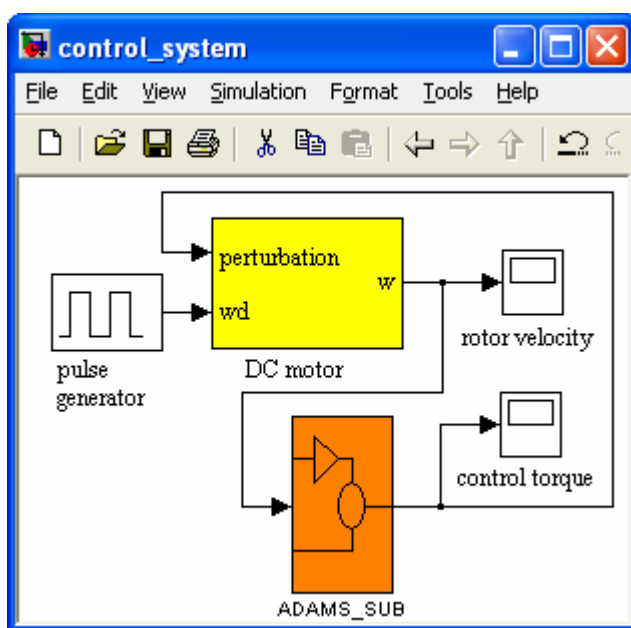


Fig. 15

For finishing the simulation diagram (fig. 15), the DC motor block was integrated with the mechanical system block (ADAMS\_SUB), and with the input signal block for the angular velocity of the rotor (Pulse Generator). For beginning, the *.m* file is loaded in the MATLAB command line, and then, for creating the MSC ADAMS block diagram, there is a special command, namely *adams\_sys*. This command builds a new model in Simulink (*adams\_sys.mdl*), which contains the MSC Software S-Function block (fig. 16), representing the non-linear MSC ADAMS model (i.e. the mechanical system). The ADAMS block is created based on the information from the *.m* file.

To set the simulation parameters (e.g. the solver type, the communication interval, the number of communications per output step, the simulation & animation model), the MSC ADAMS Plant Mask is used. In this way, ADAMS accepts the controlled input from MATLAB and integrates the mechanical model in response to them. At the same time, ADAMS provides the motor torque for MATLAB to integrate the control system model.

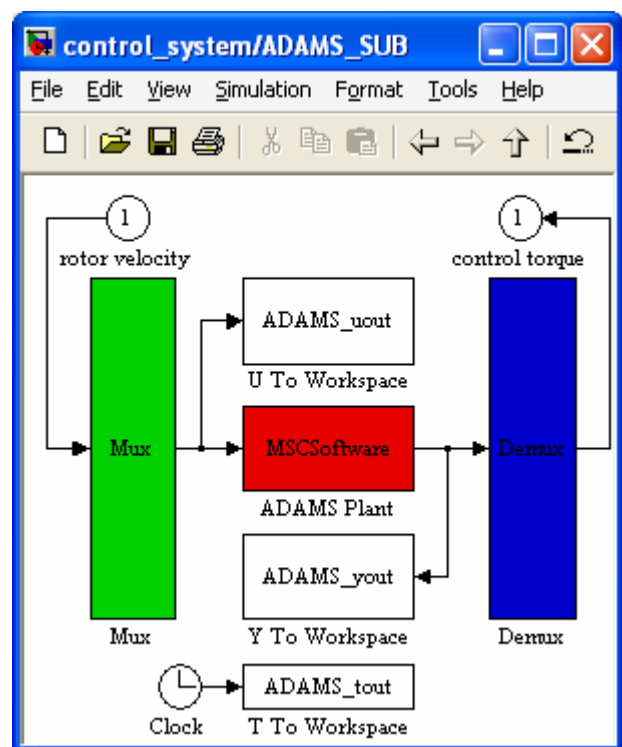


Fig. 16

## 4 Results and Conclusions

Using the above-presented design algorithm, for the double-lever windshield wiper mechanism shown in figure 4, a lot of results were obtained in order to evaluate the dynamic behavior. For example, in



figure 17 there are presented the time history variations for the input and output parameters (the control torque and the angular velocity of the rotor), while in figure 18 there are shown the variations of the current voltage and intensity. At the same time, the diagram of the reference signal is shown in figure 19. The results correspond for two complete rotations of the motor crank. As we can see, the control system generates the angular velocity of the motor crank (in the regime phase,  $\omega_1 \approx 360$  degrees/second), the bounds appearing when the lamellas change the rotation direction (the friction forces change the sign).

The application is a typical example of virtual prototyping. One of the most important advantages of this kind of simulation is the possibility to perform virtual measurements in any point and/or area, and for any parameter (ex. motion, force). This is not always possible in the real cases due to the lack of space for transducers placement or lack of appropriate transducers.

On the other hand, integrating the control system in the mechanical structure, the testing process is greatly simplified, and the risk of the control law being poorly matched to the real system is eliminated.

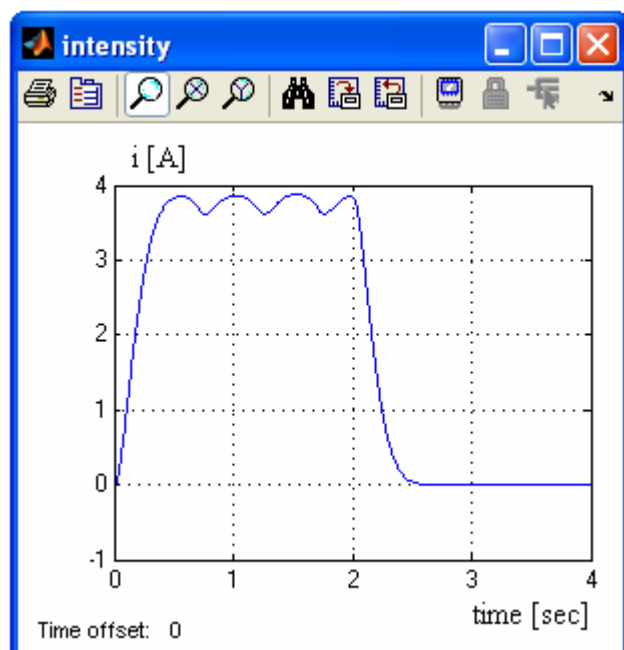
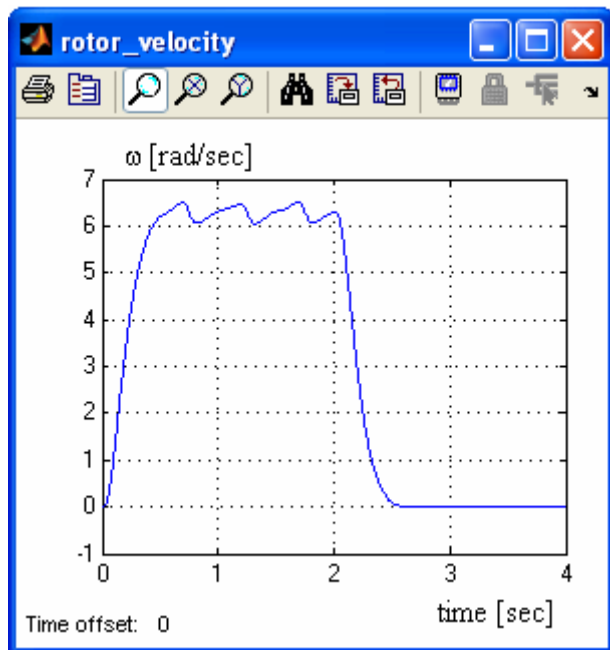
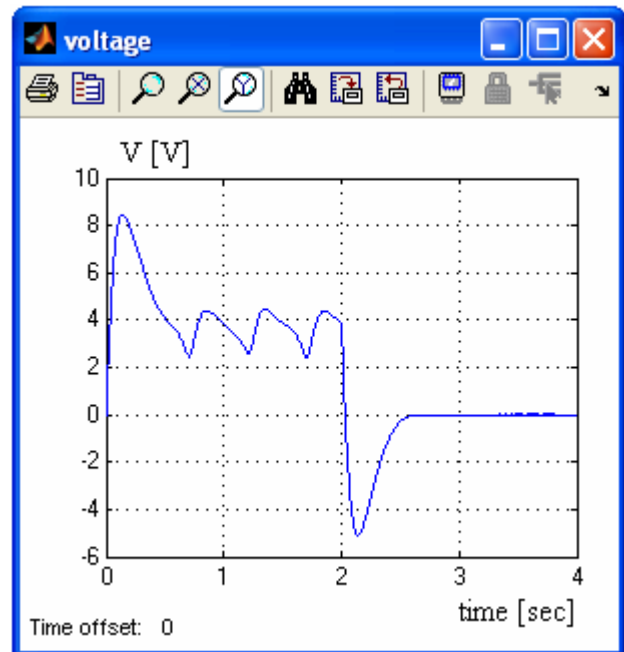
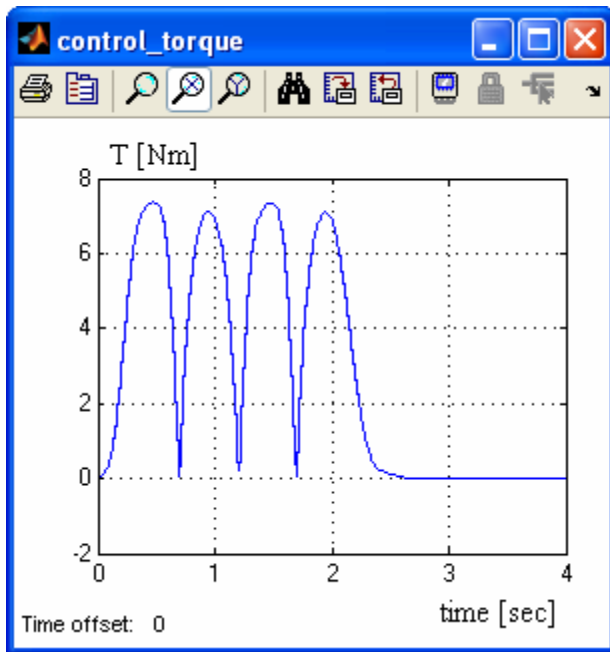


Fig. 17

Fig. 18

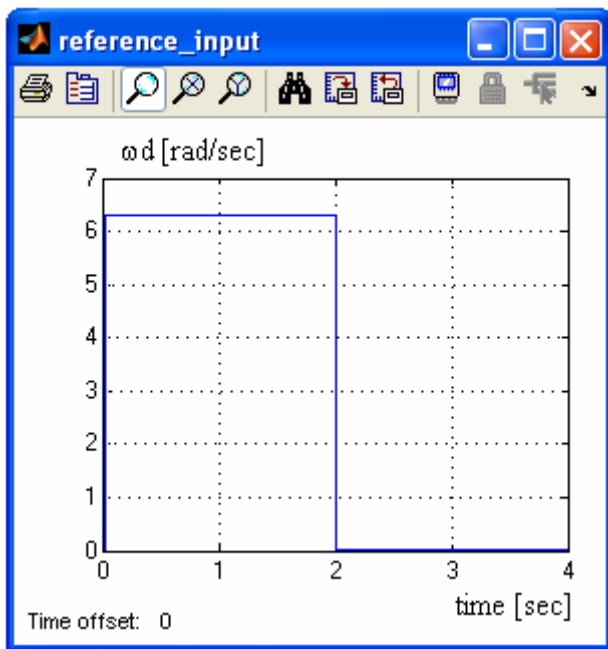


Fig. 19



Fig. 20

The windshield wiper mechanism has been manufactured, and it will be tested by using an experimental stand (fig. 20), creating a real perspective for the research in the field. This will allow a relevant comparison between the virtual prototype analysis and the data achieved by measurements, in the validation process of the virtual prototype. The results of the experimental study will be presented in a future paper.

At the same time, we intend to continue the research by modelling the mechanical structure with finite elements, for identifying the eigenshapes and eigenfrequencies of the system, which are useful to avoid the resonance phenomenon due to the action of the external dynamic loads. No less important will be to determine the maximum deformabilities, the equivalent stresses, or the Poincaré maps.

In this way, we can investigate and evaluate the complex dynamic behaviour of the automotive wiper system, without going through expensive hardware building and testing.

#### References:

- [1] D.G. Ullman, *The Mechanical Design Process*, McGraw-Hill, 1997.
- [2] L. Li, Q. Li, Vibration Analysis Based on Full Multi-Body Model for the Commercial Vehicle Suspension System, *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, 2003, pp. 203-207.
- [3] N. Nagashima, R. Nishimura, R. Ochiai, G. Fujita, Construction of Highly-Accurate Simulation Model in Automobile's Power System, *Proceedings of the 7th WSEAS International Conference on Electric Power Systems, High Voltages, Electric Machines*, 2007, pp. 334-340.
- [4] S. Vlase, H. Teodorescu, L. Scutaru, On the Form of the Motion Equations of the Multibody Systems with Elastic Elements, *WSEAS Transactions on Systems*, Vol. 6, No. 1, 2007, pp. 190-193.
- [5] C. Alexandru, I. Barbu, Modeling the Windshield Wiper Mechanisms as Multi-Body Systems, *Proceedings of the IX IFTOMM International Conference*, 2004, pp. 29-34.
- [6] E. J. Haug, K. K. Choi, J.G. Kuhl, J.D. Wargo, Virtual Prototyping Simulation for Design of Mechanical Systems, *Journal of Mechanical Design*, Vol.117, No.63, 1995, pp. 63-70.
- [7] G. Romero, J. Maroto, M.L. Martinez, J. Felez, Training in Engineering Design using Virtual Prototypes, *WSEAS Transactions on Computers*, Vol. 5, No. 4, 2006, pp. 753-762.
- [8] Q. Shen, J. Gausemeier, M. Grafe, Virtual Prototyping: To Compose Mechatronic Systems in VR Environment, *WSEAS Transactions on Computers*, Vol.5, No.4, 2006, pp. 727-734.
- [9] R. Dorf, R. Bishop, *Modern Control Systems*, Prentice Hall, 2001.
- [10] Z. Affi, L. Romdhane. ADAMS/SIMULINK Interface for Dynamic Modelling and Control Of Closed Loop Mechanisms, *Proceedings of the 7th WSEAS International Conference on Automatic Control, Modeling and Simulation - ACMS*, 2005, pp. 353-356.
- [11] K. Gunnar, B. Vladislav, The Simulation of Electromechanical Drive with DC Motor, *Proceedings of the 5th WSEAS International Conference on Power Engineering Systems - ICOPES*, 2005, pp.145-150.