Predesign of Automotive Independent Suspensions: Implementation as MDesign Calculation Module

COVACIU Dinu[1, a *], PREDA Ion [1,b], DIMA Dragoş-Sorin [1,c] and CHIRU Anghel[1,d]

[1]Automotive and Transport Department, Transilvania University of Brasov, Blvd. Eroilor, no. 29, 500036 Brasov, Romania

[a]dinu.covaciu@unitbv.ro, [b]pion@unitbv.ro, [c]d.dima@unitbv.ro, [d]achiru@unitbv.ro

Keywords: automotive suspension, design algorithm, software implementation.

Abstract: Implementation of an algorithm as a software module is not an easy task for a mechanical engineer. Usually the design engineer needs to be assisted by a programmer when a new software module is to be designed. The MDesign approach offers a way to create custom calculation modules by the user himself, who in this case is the automotive engineer, using only basic programming knowledge. This paper is focused on the development of a computation software module when the algorithm is already established. It is presented the case of the independent suspension predesign algorithm, but the same way is used for any other mechanical system. The MDesign framework was created for mechanical design and it was extended with a custom component dedicated to automotive engineering. A calculation module consists of program code and data, and is designed to execute an implemented algorithm and to provide the user with some text and graphics information.

Introduction

Mechanical design is one of the activity that benefits most from the evolution of information technology, starting with the introduction of CAD (Computer Aided Design) on personal computers, and then with the development of analysis, simulation and design management software. A special attention was given to the management of documents and product data. Now all the departments involved in the design process have access to all data and documents of their enterprise, thanks to the high-end PDM (Product Data Management) software and PLM (Product Lifecycle Management) solutions [1]. The analysis and simulation tools help engineers to optimize and verify the geometry and functionality of component and systems. However, apart from being supported by a number of software tools aimed at modeling, simulation and data storage, design engineers don't get much assistance [2] - technical management hasn't realized the full potential of workflow in conjunction with information technology. That means there is still much to work in the field of computer assisted engineering, and this is not limited to analysis like finite elements method or mechanism simulations.

In the first stage of designing a new system, the engineer has to take some important decisions regarding the functionality, dimensions and material. This is the predesign phase, where the experience has an important role, and many verification calculations are done. This is also the case of the automotive suspension system. There are some different possibilities to realize the functions of the suspension. In case of a passive suspension are used elastic elements, shock absorbers and guiding mechanisms.

The algorithm implemented and presented further in this paper refers mainly to the strut-and-arm independent suspension and to the short-long A-arm independent suspension, but the application can be easily adapted for other suspension types. The model used is described in detail in [3], and here only the basic approach is presented.

The programming environment chosen is MDesign, a software package created especially to help mechanical engineers at their technical calculations. What is special at this software package is that it offers the user the possibility to create his own calculation modules, using a custom programming interface.

MDesign Environment

General Presentation

MDesign is a software package that consists in a group of software components and applications for mechanical engineering [4]. The main components of the package are:

- MDesign Author, which is an authoring tool used to develop application libraries;
- the know-how components consolidated in these application libraries;
- MDesign Explorer, which is the graphical work environment (GUI);
- the database and communication infrastructure linked to the server.

MDesign Explorer is designed for technical calculations and includes many catalogue functions. It offers also access to databases and technical documents, including drawings and models.

The user interface has the common functionality of Windows applications, having the menus and toolbars intuitive and easy to find. There main user interface has some dedicated areas:

- an area for inputs, where the user add entry values for input variables of the calculation;
- an area for results, where are displayed the calculation results;
- an area for text help, where can be found explanations about how the calculation is performed, calculation steps, formulas, standards or other documents;
- an area for graphic help, where can be displayed graphics in various formats. The same area can be used to plot diagrams resulted from the calculation.

Fig. 1 shows the user interface of MDesign when a calculation is running. The four main areas can be easily observed: the input and output areas above the text and graphic help areas.

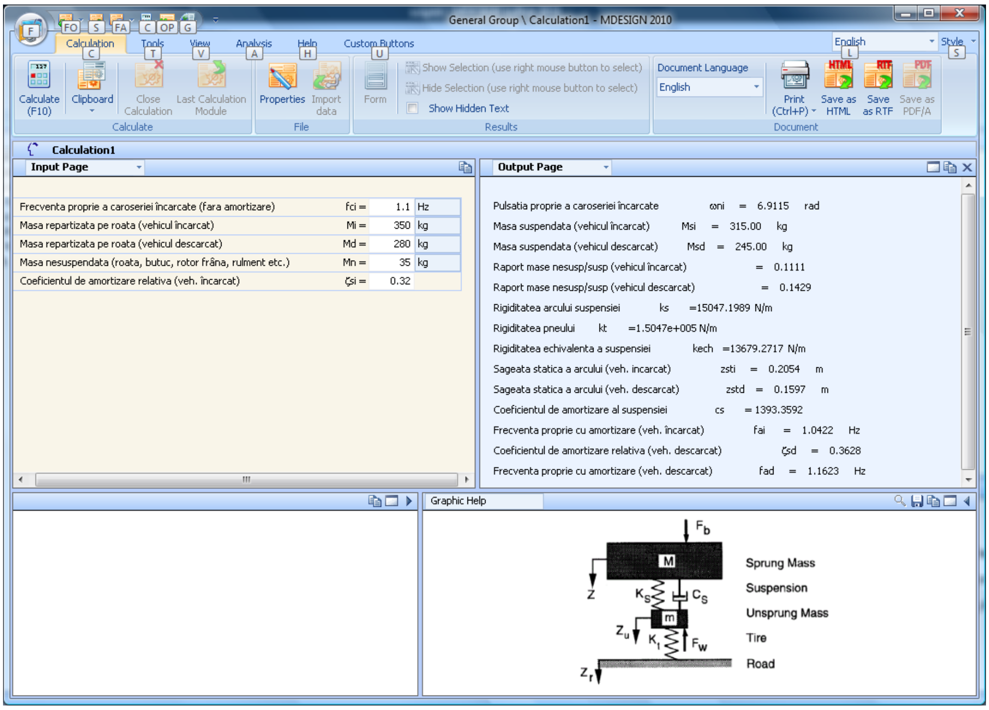
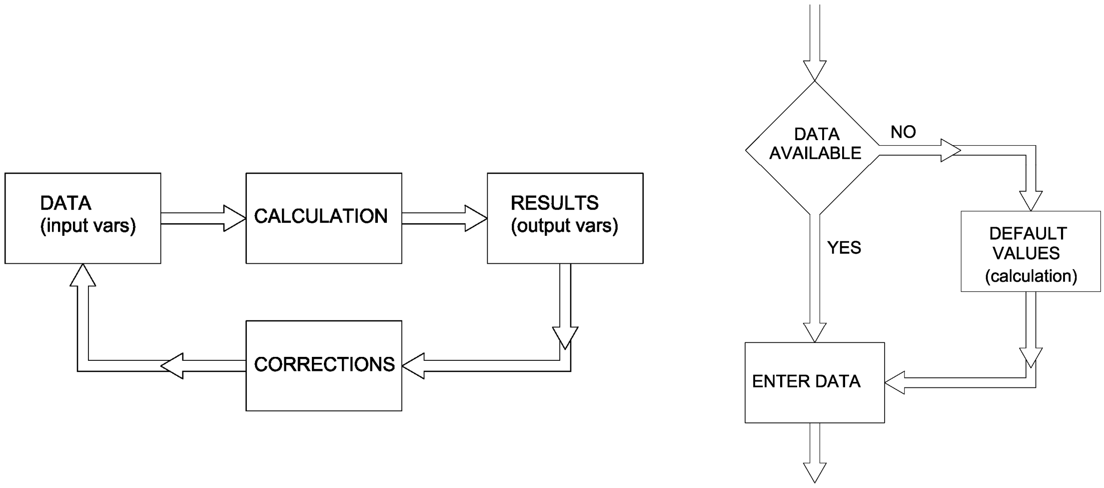


Fig. 1. The user interface of MDesign when running a calculation

The input and output areas of the user interface are not displayed sequentially. Both areas are visible on the screen at the same time and the user can change any input data, then press again the "calculation" button to see the new results in the output area of the screen. This way, a feedback loop is implemented, as shown in the diagram in Fig. 2 (a). This functionality is already implemented in MDesign basic framework, so the programmer doesn't need to spend time with implementing a user friendly interface. And the interface is the same for any other calculation modules, so the user doesn't have to accommodate with the software for any different calculation.



(a) (b)

Fig. 2. General diagram of one MDesign calculation module (a) and the flow diagram of a data input decision (b)

MDesign Author

The authoring tool is a separate part of the MDesign environment and it is not mandatory for the functioning of the already existent calculation modules. On the other hand, authoring tool needs the explorer component in order to run a newly developed module (when the "Run" button is pressed in the programming environment, it is automatically launched the Explorer). Although the possibilities are wide and very complex applications can be created, the basic functionality is very simple and includes three steps: define input variables, define output variables and write the code for calculation of each output variable.

All calculations carried out using this tool have a unique ergonomic interface for the ultimate user. The programmer (the author of calculations) receives a "comfortable development environment of input templates, output sheets, calculation algorithms and help information" [5]. On this way, as stated in the MDesign Author programming manual, a calculation procedure is developed, carefully documented and ready to be used directly.

As in any computer program, some decisions need to be taken during calculation, which need to be implemented in the software. A typical example is illustrated in Fig. 2 (b) and is dealing with the availability of the input data. There are situations where input data are available, but sometimes those data are not available and have to be assumed using default values or default calculation (like the unsprung mass in the case of the automotive suspension). So, not only variables and operators are used, but also conditional instructions (*if*). These lead to more complex instructions like loops (*for*, *while*); also complex data types, like arrays and structures can be declared; in other words, it is a programming interface inside the engineering calculation tool.

MDesign Author uses a proprietary programming language, which is derived from C++ language, but the syntax is closer to Pascal, which is simpler and more appropriate for non-programmer developers. The user friendly interface of MDesign Author is shown in Fig. 3. As can be seen, the text and graphic help areas are presented also in this interface, and a special button (with caption "Project information") allows the user to configure the content of these areas. The graphic help area can be changed also by program, so it is possible to create custom drawings during the program execution, like the diagrams created with our application and presented further in this paper.

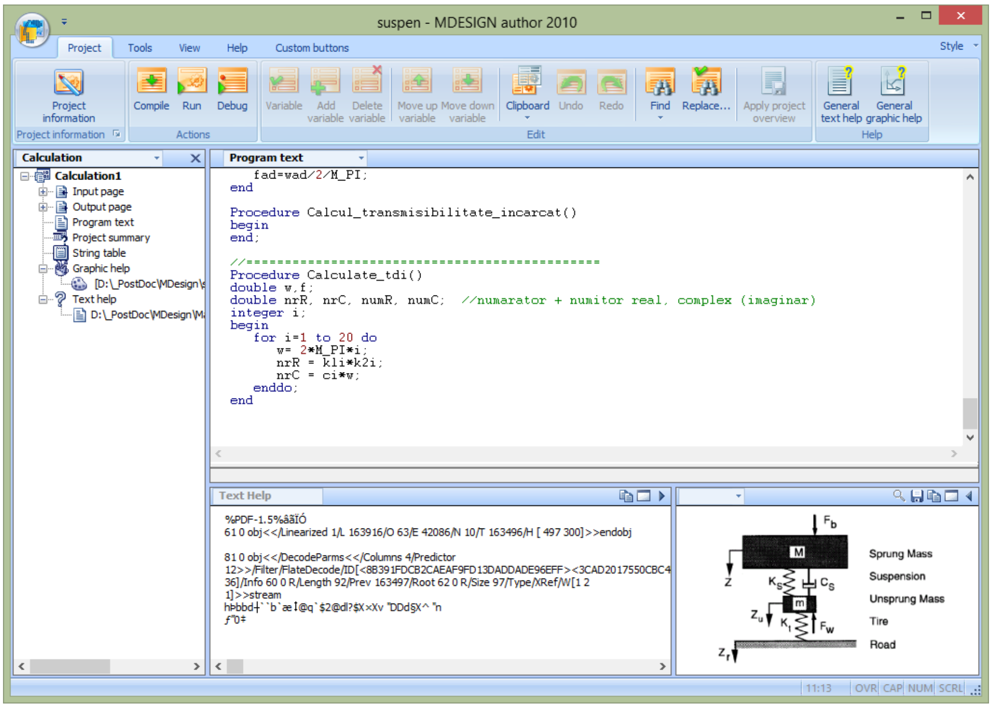


Fig. 3. MDesign Author interface

Implementation of the suspension predesign algorithm

The dynamic behavior of the vehicle can be characterized by input-output relationships. As explained in [3], the input may be the force induced by the irregularities of the road surface. The output will be the vibrations of the car body. The ratio of output and input amplitudes represents a "gain" for the dynamic system, known as "transmissibility", a non-dimensional ratio of response amplitude to excitation amplitude for a system in steady-state forced vibration. The ratio may be one of forces, displacements, velocities or accelerations.

The model of the suspension used is presented in Fig. 4, according also to [3] and [6]. The body (sprung mass) is linked to the wheel (unsprung mass) through a system composed by spring and damper without mass, while the tire (also a spring-damper system) makes the connection with the ground. The sprung mass and the unsprung mass are input data, as well as the natural frequency of the body. The output variables include the stiffness of the spring, the static deflection of the spring, the damping coefficient of the suspension and finally the transmissibility (of accelerations) for various situations.

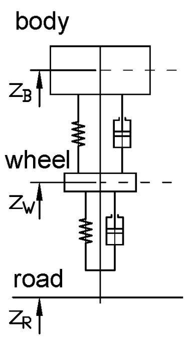


Fig. 4. Basic suspension model [3]

The algorithm starts from the Newton's second law applied to the sprung and unsprung masses, so the system’s equations of motion will be obtained. Considering the vehicle at rest, the force produced by the spring (between body and wheel) can be expressed as:

$F_{1S} = k_S \cdot d_{Sst} = W_B = m_B \cdot g$ (1)

where k_S is the spring stiffness, d_S is the spring displacement (static in this case), while W_B is the body weight and, consequently, m_B is the body mass and g is the gravity acceleration. All these are applied to a quarter of the vehicle (with four wheels).

On the other hand, the natural frequency of the body is:

$$f_B = \frac{1}{2\pi}\sqrt{\frac{k_S}{m_B}} \quad (2)$$

Starting from Eq.2, the stiffness of the spring can be calculated, as the natural frequency of the body is adopted, according to the literature (1 - 1.5 Hz, according to [6]). It is not the goal of this paper to present the algorithm in detail, but to show how this is implemented as a software calculation module. In MDesign Author, the input variables (the masses, the natural frequency) are defined in the user interface, as real numbers, without the need to write explicit code. The user has also the possibility to choose the type of data and the units (for example, if the input variable represents a force, the user can choose the unit as N, kN or other derived from these). When defined the output variable (k_S), the programming environment will also define automatically a calculation procedure. The user (programmer) has to input the code of the calculation procedure, like in this example:

```
Procedure Calculate_ks()
begin
    wni= M_PI*2*fci;
    ks = Msi*wni*wni;
end
```

where the name of variables corresponding to the input/output values as per Eq.1 and Eq.2 are: *fci* for f_B, *Msi* for m_B and M_PI is the constant π.

Normally, the masses are input data, but if the unsprung mass is not known, it can be adopted as percentage of (in example is 10%) the whole mass (on that wheel), and the calculation procedure will follow the diagram in Fig. 2 (b).

Similar functions are developed for the others output variables, and finally the transmissibility functions are calculated. These are implemented using a *for* loop. Inside this loop is called a graphic procedure that draws a line between two points (the coordinates are given by the values of the current frequency and the value calculated for the transmissibility function). Before (or after) the loop is called a dedicated procedure that define the general layout of the diagram, with axes and text and so on (see Fig. 5 - the diagram in figure is exactly as it is generated by the software function).

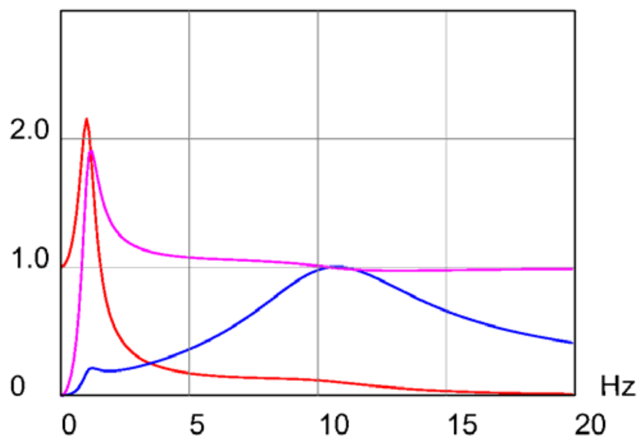


Fig. 5. Resulted diagram for accelerations transmissibility functions (non-dimensional): red - road-body; blue - wheel-body; magenta - body-body

This is only a part of the whole calculation. The output variables defined in this first part are used then in the calculation of spring and damper.

Conclusion

An MDesign module is a unit consisting of program and data, designed to carry out a calculation or another algorithm and to be able to provide the user with text and graphics information [5]. Using the authoring tool of the MDesign package, a custom module was created, to assist the engineers with predesign of an automotive suspension. Unlike other software development tools, MDesign Author can add some limit values for input variables when these are declared, using a friendly dialog box. So the programmer don't need to take care about all the error treatments needed. Also the units of measure are predefined, helping the final user to add correct values in the input area. The graphic functions are simple, but it is possible to create schematic drawings and diagrams, at run time. The presented example does not cover all the possible situations, but can be extended for any new case, as needed. This module is intented to be a part of a comprehensive library dedicated to automotive systems - an useful tool for design engineers and also for students.

Acknowledgement

This work was partially supported by the strategic grant POSDRU/159/1.5/S/137070 (2014) of the Ministry of Labor, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013.

References

[1] D. Covaciu, Study of the dynamic and in-traffic vehicle behaviour using GPS and CAD applications, Ph.D. Thesis, Transilvania University of Brasov, 2010.

[2] W. Gruender, Holistic Approach in Automotive and Mechanical Design Requires Structured but Flexible Workflows, International Congress on Automotive and Transport Engineering CONAT 2010 Brasov (2010).

[3] I. Preda, Model Based Algorithm for the Study of the Vehicle Suspension, Applied Mechanics and Materials, 2015.

[4] Tedata GmbH, MDesign User Guide, Bochum, 2010.

[5] Tedata GmbH, MDesign Author User & Programming Guide, Bochum, 2010.

[6] T.D. Gillespie, Fundamentals of Vehicle Dynamics, Warrendale, USA, SAE, 1992.