A Modular Approach to Web Based Multibody Dynamic Simulation

T. Larsson, A. Larsson, L. Karlsson
Division of Computer Aided Design
Department of Mechanical Engineering
Luleå University of Technology, Luleå, Sweden

Abstract
Computer-based tools for modelling and simulation have changed the best practise of product development. Simulation of mechanical dynamic systems have a large potential in product development but are only partly used today due to, for example, modelling complexity. A method, or tool, that supports distribution of multibody dynamic analysis models, in a modular way, is proposed and developed. Ethnographic methods have been used as a means for gaining an understanding of the engineering analysis work practice. The tool incorporates the engineering simulation packages ADAMS and MATLAB in a web based environment, and allows distributed multibody dynamic simulation in product development.

Keywords: Dynamic analysis, Distributed mechanical engineering, Simulation module

1 INTRODUCTION

The last decade's rapid development and implementation of computer aided design (CAD), or computer aided engineering (CAE), tools have together with the implementation of Concurrent Engineering (CE) changed the best practise in product development. The CAE tools is a key technology in CE [1]. CE is recognized by integrated communication, data handling and software standards. With the term CAE, one implies the engineering disciplines, and software packages, available to carry out the needed computational efforts in engineering design.

Modelling and simulation of complex mechanical engineering systems, such as multibody systems (MBS), have a large potential [2]. Mechanical systems often require restructuring, or decomposition, into subsystems [3,4] to make the system treatable for the engineering simulation teams. This design philosophy however, creates a number of problems when it comes to simulation of the overall system performance. This since the system performance can only be tested very late in the development process when the subsystems are combined into a full system again, leading to an uncertainty for the development period [5]. Although, if subsystem interfaces are set, the subsystems can be modelled and treated independently of each other. Parallel development and optimisation of modules is possible. The modular approach makes it possible for engineers to work with the part of the complex system they have engineering expertise in. The assembly and simulation of the, hierarchically seen, top simulation model concerns coupling the simulation sub modules together. The coupling of the subsystem models can be made at three different levels of model descriptions [4]: physical, mathematical or behavioural.

The engineering system is in the physical description domain represented by a physical model. The mathematical description is represented by the equations of motion. The numerical simulation results, i.e. body positions and velocities, represent the behavioural model description.

Research in coupling on all three levels, i.e. standardisation efforts, is ongoing work. STEP [6] efforts include multibody system data and standardisation in the multiphysical domain. MechaSTEP [7] include the integration of mechanics into interdisciplinary simulation. In the mathematical model description the hardware description language VHDL-AMS [8,9] is one effort. The equation oriented simulation software DYMOLA [10,11] is another. Processed DYMOLA models results in equations, solvable with own solver DYMOSIM or exported into ACSL (Advanced Continuous Simulation Language), MATLAB [12], C or Fortran source code. MODELICA [13] is an effort in standardising the modelling of multidomain systems, creating a uniform modelling language. The simulation coupling discussed in this paper concerns the physical level with connections to the mathematical description level.

The idea of distributing computer-aided design services in a computer-network, or Internet based, design environment has been addressed by several authors [14,15,16].

In this work a collaborative MBS simulation environment, with its application within vehicle system dynamics is proposed and developed. Based on the belief that events can be correctly understood only if observed in their natural environment [17], ethnographic methods have been used in order to understand and describe the engineering analysis work practice that the tool is supposed to support. The simulation packages ADAMS [18] and MATLAB are used together with database and web technologies. The collaborative tool lets the user perform simulations of systems, consisting of modules that can be distributed within company departments or externally, via the Internet, to subcontractors and suppliers.

2 DESIGNING FOR USABILITY

When building a tool that is supposed to truly support engineering analysis, i.e. designing for usability, design decisions need to be rooted in the real work activities of the users – not in rigid, written specifications that, all too often, are nothing more than preconceived notions about the actual engineering work [19]. If users and system developers do not share a common practice, or a common language, communication breakdowns are bound to occur at some point in the design process, eventually forcing the users to adapt to the system instead of the other way around.

2.1 Empirical work

Qualitative research methods [17], such as participatory observation, written field notes and informal conversations have been used in order to gain an increased understanding of the engineering analysis work practice that the tool is supposed to support. In this sense, *qualitative research* means that our aim was to understand user perspectives and needs by studying their actions in their natural work environment, as opposed to gaining requirements mainly through interviews and questionnaires. The subjects of our ethnographic study are mechanical engineers with differing skills and experience within the field of dynamic analysis. The overall aim of the study was to find out how engineers *really* work with tools for dynamic analysis, and how the computer tools that they have chosen to use support their work activities. Since the ADAMS software package is a widely known and used tool for dynamic analysis, we concentrated on studying ADAMS users.

Our empirical work shows that the activity of dynamic analysis is not just about humans and computers exchanging unambiguous information through input and output. Essentially, dynamic analysis means creating a meaningful interaction, where the computer guides the engineer through a complex translation phase - going from internal and external models to a virtual representation of the product being developed.

2.2 Work-of-the-work vs. Work-of-the-tool

Performing dynamic analysis also involves activities, which easily could be defined as bottlenecks in product development. The engineers are exposed to constant strain, which more often is related to tool-oriented problem solving, i.e. the software, than to work-oriented problem solving, i.e. engineering analysis. One of the ADAMS experts said that he and his colleagues often talk about *pushing through* or *sweating out* an analysis, which highlights that performing advanced dynamic analysis is generally considered a very difficult and time-consuming process. To put it another way; *tool-related problems are so common that it is hard to tell the difference between the work-of-the-work and the work-of-the-tool.*

2.3 Ethnographic findings

When designing a simulation supporting system, there are a couple of main findings to consider. These findings are divided into three categories – *input*, *output* and *feedback* - in order to emphasize the fact that engineers are dealing with bottleneck issues at several stages of the analysis activity.

Input

- *Information overload* occurs when the user has to consider multiple options for input, some of which are completely irrelevant for the task at hand, and some of which the user has no knowledge about.

- There is a lack of *natural mapping* between the way the user wants to do things, and the options and sequence of steps that the system suggests.

Output

- *Information overload* occurs when the user has to take multiple answers into account, some of which are completely irrelevant for the task at hand, and some of which the user has no knowledge about.

- There is lack of *natural mapping* between the way the user wants to have the results, and the way that the system provides the result.

Feedback

- The way that the studied system provides feedback is not efficient enough. The users still have to apply a *trial and error* approach to get useful feedback. There is a lack of both real-time feedback and feedback which provide not only warnings or errors, but also suggest ways to deal with these problems.

The system forces the users to let go of the main thread - stepping out of their *engineering mindset* - only to handle the tool, which actually was intended to support their engineering work. They are continuously using parts of their intellectual capacity to reason about and act according to events, which do not really add value to the ongoing process of product development.

Presumably, thought and reflection are more powerful when applied to problems that are relevant to the main activity, which, in this case, is *performing dynamic analysis*, not *using tools* for performing dynamic analysis.

2.4 Design Rationale

One possible way to approach the above issues is to help the user understand the underlying reasons for design, i.e. assisting the user in understanding the functionality of the tool as well as how to use the simulation model. In this respect, we are also designing to facilitate learning. Since the user is not an expert in performing analysis, there are obvious advantages with an application that can mediate some of the knowledge and expertise that the simulation expert possesses. A distributed system for dynamic analysis is not just about distribution in time and space; it is also about distribution of knowledge.

Design rationale [20] is a concept that on many accounts represents this view on *knowledge distribution*. It is an excellent way to give the users access to a richer view of the tool, and thereby also a richer view of how to use it.

Even though design rationale traditionally is used as design documentation, to be used by other designers, a similar technique can be used when creating a meaningful interaction between the user and the analysis tool. Incorporating design rationale into the distributed MBS system will help the users to an increased understanding of the tool and its functionality, since the design intent of the simulation expert/model builder is communicated, including the underlying reasons for choosing certain parameters or ranges, and any kind of information that could assist the domain expert in performing the analysis.

3 WEB BASED MODULAR SIMULATION TOOL

The collaborative MBS environment is based on the idea of simulation software experts building models within the simulation software and sharing these models together with a model specification file. The end user can choose simulation models, change sub model parameters with help of design rationale, and finally perform analysis and interpret results. The server side software performs total simulation model assembly, simulation and result generation.

3.1 MBS application

A multibody system is a system that consists of solid bodies that are connected to each other by joints that restrict their relative motion. The study of multibody dynamics is the analysis of how mechanism systems move under the influence of forces. The MBS analysis methodology for performing analysis in the product development process is seen in Figure 1. Simulation processes are carried out at different stages of the development and with different types of objectives and with different level of detail of the simulation models.

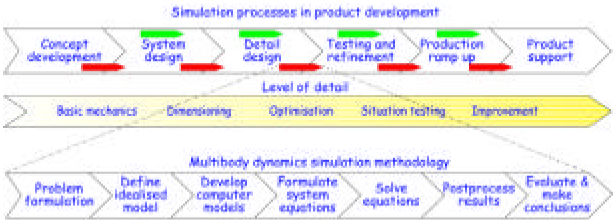

Figure 1: MBS analysis methodology in PD.

The vehicle system application in this work is a front suspension of a car, a quarter model. The simulation model consists of elements that are to be treated as modular. These elements are, for example, the shock absorber, the control for active shock absorber, the road profile, the tyre and the steering control. The simulation models are built in ADAMS and MATLAB. The physical representation of a system configuration is shown in Figure 2.

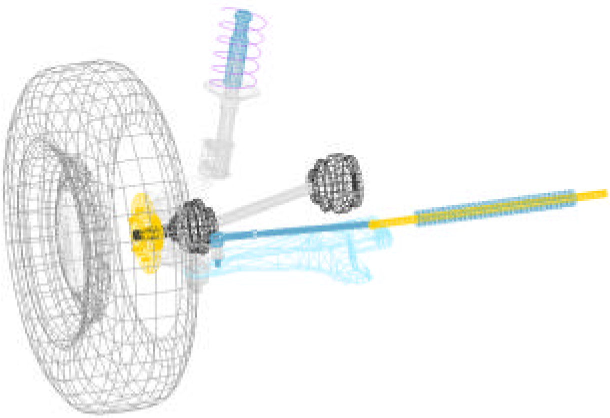

Figure 2: Modular quarter model of automotive front suspension.

The motive for analysis of a mechanical system such as a vehicle suspension is the prediction of ride comfort, handling and stability [21]. One of the goals is to find an optimal parameter set for prescribed workloads, such as passing over a cleat, asymmetric or symmetric pothole, and J-turn manoeuvre.

3.2 Modelling procedure

When building the models to be used by the distributed system there are concerns with *global model structure*, *subsystem modular structure* and *simulation module coupling*. All simulation models are built in ADAMS and MATLAB. The ADAMS models are 3D models with physical properties derived from I-DEAS [22] solid models. The control models are built in MATLAB. The procedure for building the simulation models could be according to Figure 1.

Global structure

When creating the global structure this includes the definition of the top model i.e. the world of reference for the environment. In this case the world of reference contains gravity and a world reference point and ground. This is properties that span over the entire modular structure and is to be considered as the outer boundaries of the system. The different simulation modules are then implemented in the global model. The global model is exported to a text based ADAMS command file. A specification file containing parameters allowed for change, parameter standard values, parameter ranges and comments are also saved as text file.

In the global structure the adoption to the product is also made, i.e. the suspension system. The global structure is here designed to contain the different modules mentioned earlier. This implies that pointers are made to specific building blocks, or objects, in the specification file. The definition of how to connect the objects is also designed at this level.

Subsystem modular structure

The modelling of sub models is about getting a resemblance between the actual system and the modelled system. The subsystem is exported to an ADAMS command file as well. The specification file contains information like the global file, but also additional information such as input/output ports to the subsystem parameters. Modelling of control systems is performed in MATLAB where model information is stored in MATLAB text files. The linking to the ADAMS environment is performed with ADAMS macros, i.e. a sequence of ADAMS commands. Model validation by experiments and computer simulations [23] is desired to ensure parameter validity ranges. Different model generations concerning level of detail can be made. However, different modules due to different parameter values are not needed since this procedure is performed via the web interface.

Simulation module coupling

The coupling of the different simulation modules is performed via ADAMS macros, or C/FORTRAN code segments used to connect the objects according to their specifications in the object files.

Model registration

The place where the model is located, local server, remote server or web page, is registered in the database for usage by the simulation system. Corresponding specification files are supposed to be placed in the corresponding place.

3.3 Modular simulation procedure

When the underlying simulation modules are available, the user i.e. anyone with access to the web page, seen in Figure 3, can perform simulations with the available models. The client can choose composition of simulation modules, load existing set of modules, change parameters in modules, or replace modules by newly developed ones. Throughout the procedure the user is guided by design rationale supplied by the model builder, represented by **i**, **?** and **!** in Figure 3 . The user can store comments about parameter changes, for later use, or for guidance to others. To simulate, desired parameters are changed, results wanted are checked, simulation parameters such as time and steps are given and the analysis is submitted.

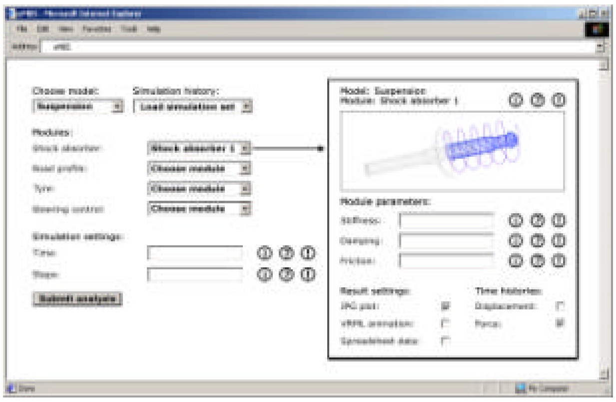

Figure 3: Web input interface.

The results can appear in a web page according to Figure 4. The possibility to perform MBS simulations is now granted anyone with access to the web page.

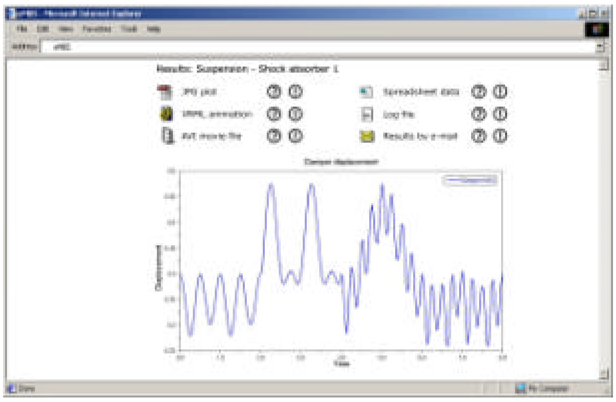

Figure 4: Web simulation results interface.

3.4 Software tool integration

The simulation system, a co-simulation system [24], is built of Active Server Pages (ASP), the simulation software packages ADAMS and MATLAB, a database and a web server, according to Figure 5. Additional development of Windows and UNIX scripts has also been made to allow both UNIX and Windows web server implementation.

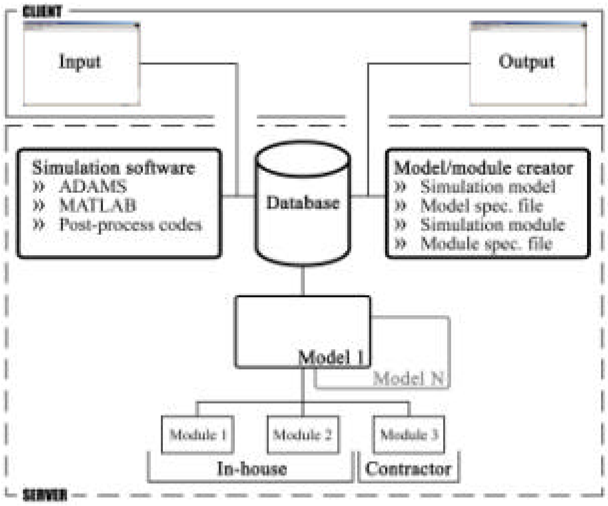

Figure 5: Tool integration.

The ADAMS software is used for simulation of dynamics in the mechanical. The MATLAB software is used for control system modelling and some plotting procedures. The two packages are coupled together via the use of text files and macro scripting in ADAMS.

The database is populated with simulation models and specification files to the corresponding models. The specification files contains information used for the ASP web pages in order to perform the routines of module selections, viewing of design rationale, parameter values, parameter comments etc. The DB keeps track of where models and model information are stored, locally or remote. Any modification made to the model via the ASP web page is stored with association to the model files.

The ASP request information concerning available modules and their connectivity possibilities from the DB. When information is chosen the DB publishes the information to the web page. When the analysis is executed the ADAMS/MATLAB environment is addressed form the ASP pages with simulation parameters and the DB releases the model. Simulation results are tracked, organised and stored locally by the DB. Generation of plots, VRML animations, AVI movies and spreadsheet data is performed by subroutines implemented in the ADAMS/MATLAB environment. The subroutines are programmed in ADAMS macro language, as MATLAB files and as Fortran and C routines.

When the analysis is complete, the output ASP page addresses the DB for simulation result information. Links to locally stored result files are published to the web page and the user can choose to download the results

The key to the working modular system is the mapping of web input parameters to simulation parameters. This mapping is made in the model specification files and gives the constraints for simulation module coupling as well as parameter coupling. Another important implementation is the ASP connection to the execution of simulation models, allowing simulation in a total scripted batch mode.

3.5 Web interface

Web interfaces require much less bandwidth than other technologies for presenting graphical user interfaces (GUI) remotely, i.e. X window systems. This enables the distributed use to include clients with low bandwidth. Web browsers are relatively mature client applications, available for almost all computer systems, and together with the use of technologies, such as Java, Active Server Pages (ASP), or Dynamic HTML, interactive pages can be developed granting access to whatever application is needed. The web interface for the simulation system is developed in ASP and DHTML, giving interactivity to the interface.

Input interface

The input interface, Figure 3, consists of a start page where simulation top model is chosen; the underlying sub model configuration is displayed with distinction between the different models parameters. Simulation sets, parameters for the chosen model/modules, settings for the simulation, such as time and simulation steps and result settings. Simulation model and modules preview is also available to provide the user with a visual overview of the system and its modules. To each parameter in the model there are additional information available, implemented by the model creator; design rationale, parameter information and comments. The number of parameters and results settings is depending on loaded model and hereby controlled by the model creator. After parameters and settings are chosen the model will be executed at the click of the submit button.

Output interface

The output interface, Figure 4, consists of results and analysis related information such as log files together with design rationale and comments. Result plots and VRML (Virtual Reality Modelling Language)/AVI animations are shown at the bottom of the page when requested. All results are stored at the server side until requested by the client, either by hyperlink or by request for e-mailed results.

4 DISCUSSION

By designing simulation tools that are closely related to the work practice of the users, and by actively involving these users in the cooperative design of such a system, a work situation where engineers can concentrate on their primary work activities is created. Focus shifts from the software system problems to the engineering analysis procedure.

The approach of simulation tool experts developing the models and domain experts performing simulation of modules within their own domain creates a structure for using the engineering expertise in a preferred way. Engineers that traditionally have had to rely on simulation experts are given the possibility of performing parts of MBS analysis themselves. Basically, they will be able to make relevant dynamic analysis earlier and more often in the process, which ultimately will leave them in the advantageous situation of increasing product quality at the same time as the time-to-market is reduced.

The system is hardware independent for the end user – only a web browser is needed - and the user can work anywhere in the world in a thin client fashion. The most evident advantage is the possibility of distributing simulation modules, or whole simulation models, from simulation experts to design experts and engineers. The distributed approach allows cooperative companies to protect company specific knowledge by distributing simulation modules of models or complete systems in a black box fashion, were some parameters are available for modification, while the company sensitive parameters and structures are protected. The structure with modules allows for parallel development of the subsystem models. An updated subsystem module can replace an older one and simulation can be performed to see the impact on the total system.

The scenario application within vehicle dynamics is to be seen as a test bed only. The effort is not in developing suspension simulation software.

To use the system as a support in product development, guides for implementation into the existing development process must be worked out. The successful implementation would lead to a scenario where MBS simulation drives the design rather than validating the design after it has been developed.

5 CONCLUSION AND FUTURE WORK

A method, or tool, that supports distribution of multibody dynamic analysis models, in a modular way, is proposed and developed. Ethnographic methods have been used as a means for gaining a deeper understanding of the engineering analysis work practice. In order to deal with the problematic issues of engineering analysis that emerged during the empirical study – such as irrelevant input and output – as well as satisfying the need for useful, real-time feedback, the concept of design rationale is suggested and applied.

The web based collaborative environment contains the engineering simulation packages ADAMS and MATLAB, and is applied to a vehicle system dynamics application. Database technology is used together with ASP web pages to create the framework needed for the total simulation tool. The tool allows parallel development of analysis modules and facilitates the usage of black-box simulation. The tool can be used throughout the different stages of the development process, regardless of geographic location. If changes are made in the existing development process the usage of this tool can help in achieving simulation driven design rather than a simulation validated one.

Future work includes more fieldwork testing, in order to establish a sound knowledge base for further integration into company development processes. Further work would also include the incorporation of more knowledge domains, such as electronics, to facilitate the concept of mechatronic simulation. The implementation of standard product models, where these are available, instead of the simulation software native formats would be an achievement.

6 ACKNOWLEDGEMENTS

The financial support from the Polhem Laboratory, the Swedish Foundation for Strategic Research and ENDREA is gratefully acknowledged.

7 REFERENCES

[1] Haug, E. J. (ed.), 1993, Concurrent Engineering: Tools and Technologies for Mechanical System Design, NATO ASI Series, Series F: Computer and Systems Sciences, Vol. 108, Springer Verlag Berlin Heidelberg.

[2] Shiehlen, W., 1997, Multibody System Dynamics: Roots and Perspectives, Multibody System Dynamics, Vol. 1, pp. 149-188.

[3] Zeid, A., Chang, D., 1989, A Modular Computer Model for the Design of Vehicle Dynamics Control Systems, Vehicle System Dynamics, Vol. 18, pp. 201-221. Swets & Zeitlinger.

[4] Kübler, R., Shiehlen, W., 2000, Modular Simulation in Multibody System Dynamics, Multibody System Dynamics, Vol. 4, pp. 107-127, Kluwer Academic Publishers.

[5] Stensson, A., Larsson, T., Merkt, T., Schuller, J., Williams, R. A. and Mauer, L., 1999, Industry Demands on Vehicle Development – Methods and Tools, The Dynamics of Vehicles on Roads and Tracks, Vehicle System Dynamics Supplement 33, pp. 203-213, Swets & Zeitlinger.

[6] Dürr, R., Endres, M., Neerpasch U., Schiehlen, W., Witte, L., 1996, Standardisierung eines neutralen Datenformats in STEP für die Simulation mechatronischer Systeme, VDI-Berixhte 1289, VDI-Verlag, Düsseldorf, pp. 381-397,.

[7] Donges, Ch., Krasel, M., 1999, MechaSTEP-STEP Datenmodelle zur Abbildung mechatronischer Systeme, Product Data Journal, No. 5, pp. 30-34.

[8] Moser, E., Mittwollen, N., 1998, VHDL-AMS: The missing link in system design – experiments with unified modelling in automotive engineering., Design, Automation and Test in Europe (DATE'98), Proceedings, IEEE-CS.

[9] VHDL-AMS: http://www.vhdlams.com

[10] Elmqvist H., Brück D., 1995, Composite Constructs for Object-oriented Modelling, Eurosim Simulation Congress 1995, Vienna, Austria, September 11-15.

[11] DYMOLA: http://www.dynasim.se

[12] MATLAB: http://www.mathworks.com

[13] MODELICA: http://www.modelica.org

[14] Regli, W. C., 1997, Internet-enabled computer-aided design, IEEE Internet Computing, pp. 39-50.

[15] Trika, S. N., Banerjee, P., and Kashyap, R. L., 1997, Virtual reality interfaces for feature-based computer-aided design systems. Computer-Aided Design, 29(8), pp. 565-574.

[16] Han, C. S., Kunz, J. C., Law, K. H., 1999, An Internet-based distributed service architecture, Proceedings of the 1999 ASME Design Engineering Technical Conferences, DETC99/CIE-9077, September 12-15, Las Vegas, Nevada.

[17] Ely, M., 1991, Doing Qualitative Research: Circles within Circles, Falmer Press Ltd, London, England.

[18] ADAMS: http://www.adams.com.

[19] Olsen, G. R., Cutkosky, M, Tenenbaum, J. M., Gruber, T. R., 1994, Collaborative Engineering Based on Knowledge Sharing Agreements, Engineering Data Management: Integrating the Engineering Enterprise, ASME.

[20] Burge, J., 1998, Design Rationale, Draft Technical Report, Worchester Polytechnic Institute.

[21] Schalk Els, P., Van Niekerk, J. L., 1999, Dynamic Modelling of an Off-Road Vehicle for the Design of a Semi-Active, Hydropneumatic Spring-Damper System, Vehicle System Dynamics Supplement 33, pp. 566-577, Swets & Zeitlinger.

[22] I-DEAS: http:/www.sdrc.com.

[23] Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P., 1989, Design and Analysis of Computer Experiments, Statistical Science, Vol 4, No. 4, pp. 409-435.

[24] Veitl, A., Gordon, T., Van De Sand, A., Howell, M., Valasek, M., Vaculin, O. And Steinbauer, P., 1999, Methodologies for Coupling Simulation Models and Codes in Mechatronic System Analysis and Design, Vehicle System Dynamics Supplement, Vol 33, pp. 231-243, Swets & Zeitlinger.