

MODELLING, SIMULATION AND VIRTUAL INSTRUMENTATION IN MECHATRONICS

Valer DOLGA, Lia DOLGA

“Politehnica” University of Timișoara, Romania, Department of Mechatronics,

E-mail: valer.dolga@mec.upt.ro

lia.dolga@mec.upt.ro

Keywords: mechatronics, design, model, simulation, virtual instrumentation

Abstract. Development of new products on the world market or renewing the existing ones is an important task of any industrial company, required by a visible decreasing of the life cycle for these products. New technologies are developed to further improve manufactured goods and offered services, associated with the quality improvement. Successful solutions for new industrial goods properly merge electronics, software and mechanics together with a better use of information. In turn, each stage in the product lifecycle shortens. The design stage has to quickly generate the optimal solution for the new or renewed artefact. Modelling& simulation have to be correctly balanced by the experimental& constructive tests.

1. Introduction

The characteristics of any goods during the entire lifecycle are mostly determined by the decisions adopted during the early stages of this cycle. Around 70% of the total costs and of the quality properties are determined during the conceive stage, although low level of resources are implied (about 5%). Due to the errors made during the initial stages in a product lifecycle, numerous “mistakes and deviations” occur during the manufacturing stage in lots of workshops. Huge amount of money are spent to ensure the quality and to test the results of the manufacturing operations. Correctly implementing and using the concept of “mechatronics” just from the early stages in the product life, one can obtain low cost and high quality products [1], [5].

Examination and measurement were at all times “lines of attack” to gain knowledge. The emergence of the “Systems Theory” opened the route to the very important stage of “modelling”. Applying the systems theory in modelling procedures confers them systemic features of a wide generality. Operational information become input/ output data for the approached system.

Development of new or existing products on the world market is an important task of any industrial company, required by a visible decreasing of the life cycle for any of these products. New technologies have to be developed to further improve manufactured goods and offered services, associated with a continuous quality improvement. Successful solutions for new industrial goods properly merge electronics, software and mechanical engineering together with a better use of information. In turn, each stage in the product lifecycle shortens. The conceive- and design stages have to quickly generate the optimal solution for the new or renewed artefacts. Modelling& simulation have to be correctly balanced by the experimental& constructive tests.

The paper deals with the connexion of the modelling& simulation concept on one side and the experiment& virtual instrumentation on the other side. All of these activities are performed in combination and must be considered as a collection of software tools and working methods integrated together to address either single stages of the lifecycle or connect different tasks of the whole process.

2. About the concepts of “model” and “simulation” in mechatronics

Nowadays, the development of a new product frequently rejects the experiment stage. One can ask indeed why and how could the industry abandon and disclaim the experiment so often. The main reasons refer to:

- the high costs of experimental procedures,
- the injuriousness of some systems,
- the difficulties to access certain systems.

Sometimes, the experiment is simply in no way possible because the system is not yet built.

A feasible solution can be:

- to achieve a mathematical model for the approached system based on the main aspects and characteristics that could be useful and actually used in the system and to apply the known laws of physics, biology, economy, etc.;
- to analyze and simulate the mathematical model by solving the equations systems (manually or automatically). The stage is essential for the acquaintance and comprehension of a system behaviour using the results of the model behaviour.

The positive result of this concept is sustained by the reduced cost of the simulation process. However, there is a caution to be adopted: the usefulness of the simulation results basically depends upon the similarity between the simulated model and the real physical system that was modelled. In fact, achieving a right model is an art, requiring talent, ability and intuition together with experience and deep knowledge and understanding of the system running mode [2].

The model definition takes into account two principles (Figure 1):

- Knowledge and intuition about the system are already existing (white box component);
- Experimental data – input/ output – from within the system are already available (black box component).

Figure 2 outlines in simple manner the signification of the topic “modelling”.

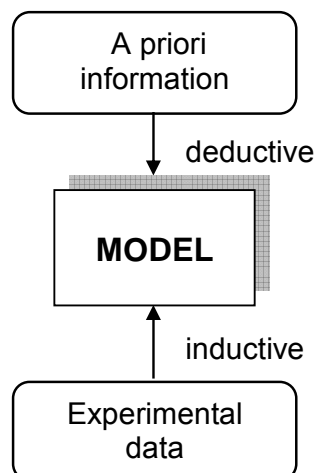


Fig. 1 The construction of a system model

The design process can be completed either in a new product/ process/ service or an improvement of an existing product/ process/ service. Appropriate product models and simulation respectively involve reducing the time for analyses and increasing the efficiency of the design activity.

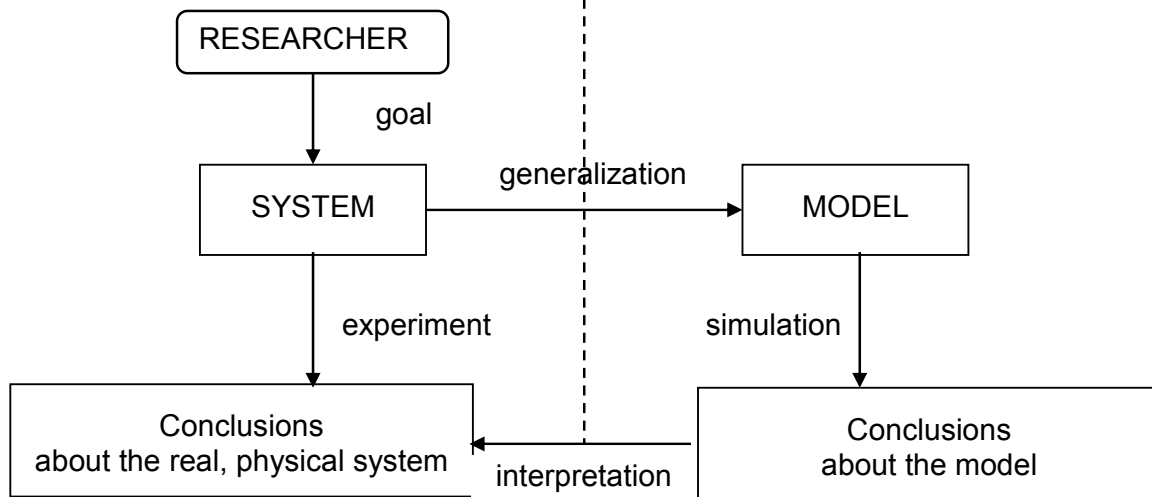


Fig. 2 The real space and the model space

Starting from the proposed goal and the definition of the question to be solved, the list of design requirements is identified and consequently, several principle solutions are generated. After selecting the working concepts, the stage of modelling/ simulation/ optimization comes in succession, so that the preliminary design can be defined, and formulated. During the modelling and simulation steps, supplementary parameters may occur [3], [4], [6].

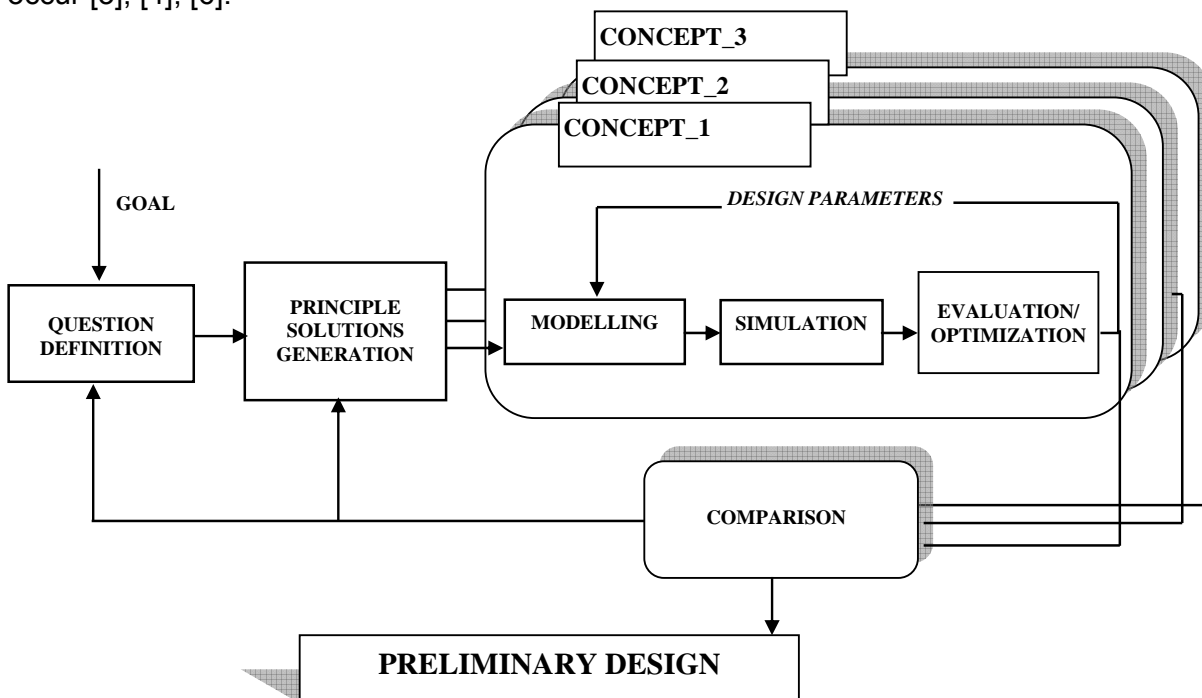


Fig. 3 The design process and the modelling stage

The essential stages of the mathematical modelling are:

- **The definition of the generalization levels:** the decisions are necessary during the first and the second succession of the modelling.
- **The selection of the description method:**

- Behaviour description: oriented on equations, differential equations (ODE);
- Structural description: the system is described using subsystems and basic (primitive) elements that are compatible to the simulator.

Many of the used equations are nonlinear differential algebraic equations (DAE).

- **The definition of the interface:** the ports are classified in **conservative ports** and **non-conservative ports**.
- **The definition of the signals properties:** these may be analogue or digital.

The significance of the topic “**simulation**” is correlated to that of “**model/modelling**” and substantially differs with respect to the application domain. The references mention many definitions for this concept; the authors decide on this one: “**an imitation process of a real phenomenon based on a set of mathematical relationships**”.

3. Symbolical and graphical programming in modelling processes

Modelling and simulation are used in obtaining certain results about different actions in a virtual environment with the purpose of evaluating these results with respect to similar actions performed in a real environment.

The high level programming languages (FORTRAN, C, C++) are used to develop models and codes for simulation. They prove an essential advantage: they do not require to the user advanced programming skills and knowledge and make available lots of tools and intuitive, rapid graphical modelling techniques.

General languages (ACSL, SIMMON, DESIRE, etc.) were brought into play with the aim of eliminating the effort to distinctly write the codes for each individual application.

Graphical programming languages are extensively used in the discussed domain. New languages were developed for various engineering fields:

- **SPICE, Electronics WorkBench** are destined to electrical and electronic circuits. They work accordingly to the topology of these circuits.
- **Working Model** is a simulator for mechanical systems centred on graphical constructions and making use of concentrated parameters.
- **LabVIEW (Laboratory Virtual Instrument Engineering Workbench)** is a programming environment based on the graphical programming language called G (Figure 4). LabView promotes and disseminates the concept of “**modular programming**”, likewise the C, C++, and PASCAL programming environments [8].

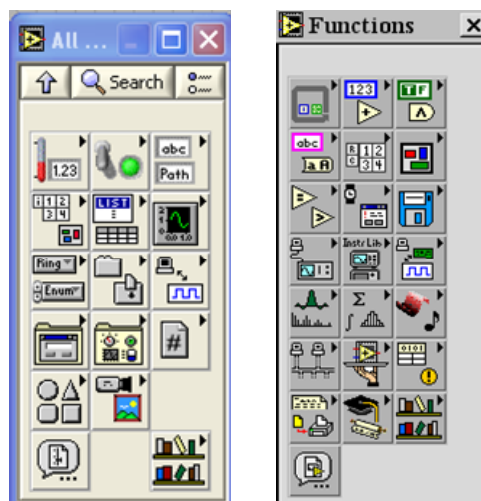


Fig. 4 Working toolboxes in LabView

LabView is a working environment conceived for virtual instrumentation (*Virtual Instruments – VI*) destined to process monitor and control. A *VI* has 3 components:

- ❖ *front panel* – corresponding to the user graphical interface, that is the image the user sees on the monitor screen;
 - ❖ *block diagram* – corresponding to the code of the program and defining the functionality of the *VI* –it uses classical operators, functions, etc;
 - ❖ *icon* and the connector corresponding to the program “signature”. The icon is the graphical identifier of the *VI*. The input and output terminals correspond to the input/ output parameters.
- **MATLAB (MATrix LABoratory)** is a high performance package of programmes dedicated to digital calculations and to special graphical representations. **Simulink** is an integrant component of this software. **Simulink** facilitates the modelling, the simulation and the dynamic analysis of the systems and is based on the graphical programming method. A block diagram editor is used. Both linear and nonlinear systems are accepted, as well as analogue and discrete systems [11].
 - A special place among the modelling and simulation languages is reserved to the **object oriented languages**. The object oriented language **Modelica** is used in modelling physical systems. It was designed to sustain the development of working libraries and the model modification. **Modelica** is an up to date language that uses mathematical equations together with the object orientation. It offers a strong library called “*Modelica Standard Library*” (Figure 5). **Modelica** is characterized by unity in its structure and makes use of entities like: *class*, *model*, *block*, *function*, *connector*, *package*, *record*, *type*. In order to be used in a simulation environment, the language **Modelica** and the library MSL require the existence of a **Modelica translator** [12].



Fig. 5 Modelica Standard Library

- **Dymola – Dynamic Modeling Laboratory** – is an environment that can be used in modelling and simulating complex systems. **Dymola** is based on Modelica® and counts on the *Modelica translator* and on a series of working libraries compatible to Modelica Standard Library: MultiBody Library, PowerTrain Library, Hydraulics Library, Pneumatics Library, VehicleDynamics Library, AirConditioning Library. **Dymola** is designed to operate with various other modelling and simulation environments. Figure 6 shows an architecture close to Dymola working availabilities is shown in Figure 6 [10].

Within the **Model level**, the models are constructed starting from the components library (Modelica standard or other commercial or individual libraries) and afterwards they are developed by the user. The detailed model can be imported from within a computer aided design environment (CAD). The exchange formats can be of DXF or STL type. One can therefore obtain information about the mass and the inertia of the 3D mechanical components and about the topology of the mobile system. The icon of the mobile component can be defined either within the graphical frame of the Dymola model or outside it, by importing the icon from another graphical environment.

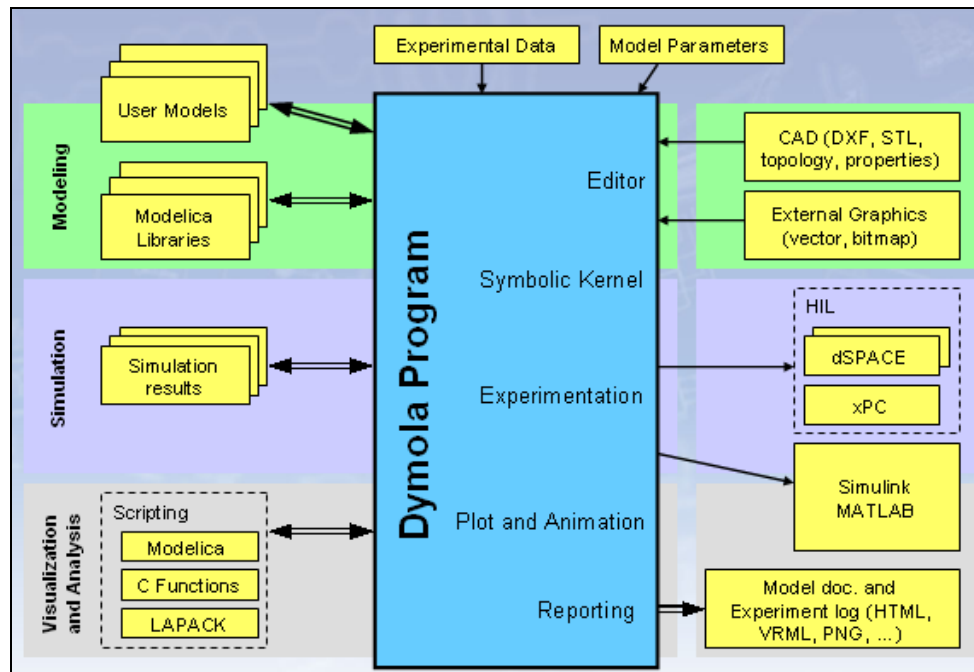


Fig. 6 The Dymola architecture

Within the **Simulation level**, Dymola transforms the described model into a simulation code. Dymola includes a complete simulation environment, but can also export the code for a Simulink simulation. Supplementary, for an off-line simulation, Dymola can generate a code for dedicated hardware: dSPACE, xPC, etc.

- **20-SIM (Twente Sim)** is an advanced program for modelling and dynamic simulation of complex systems: mechanical, electrical, hydraulic systems. This program was developed in the Control Laboratory of the University of Twente (Netherlands) and is the successor of the TUTSIM package. The model creation is based on the use of equations, block diagrams, bond-graphs, icon diagrams and combinations of the previously mentioned items (Figure 7 a...d) [9].

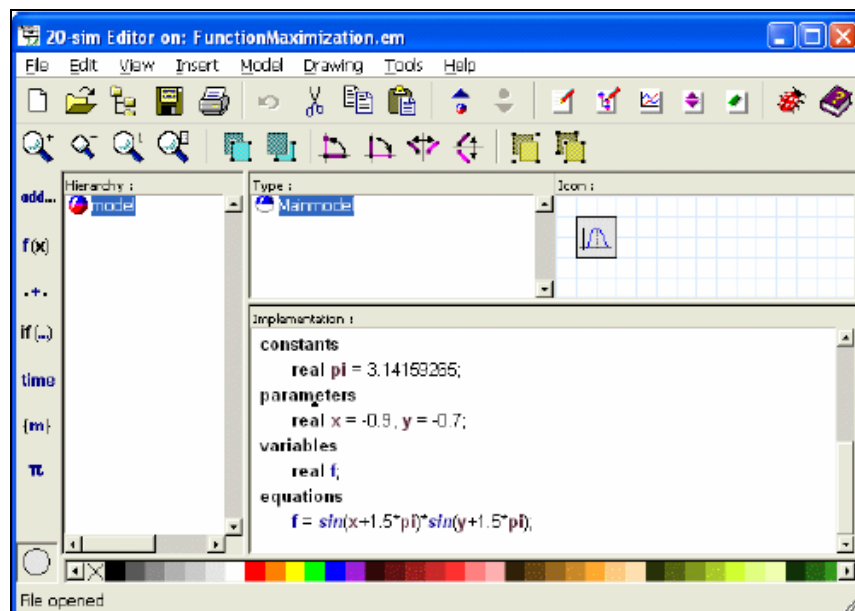


Fig. 7a A modelling example using equations within 20-SIM

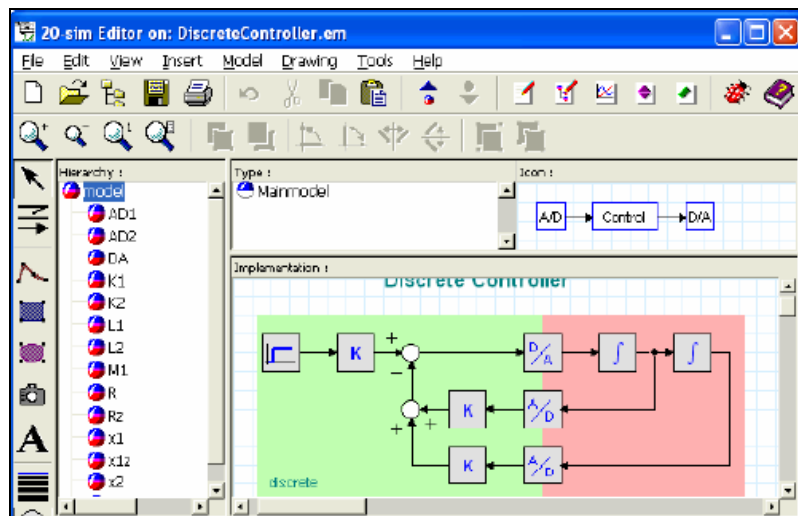


Fig. 7b A modelling example using block diagrams within 20-SIM

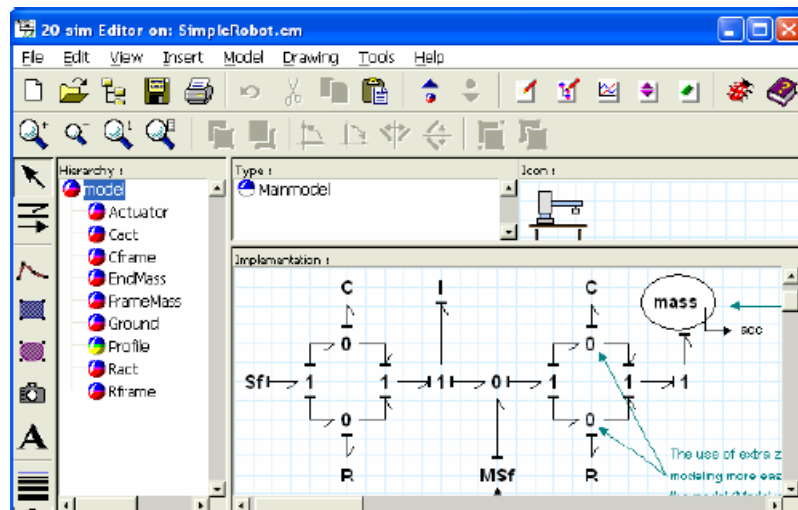


Fig. 7c A modelling example using bond-graphs within 20-SIM

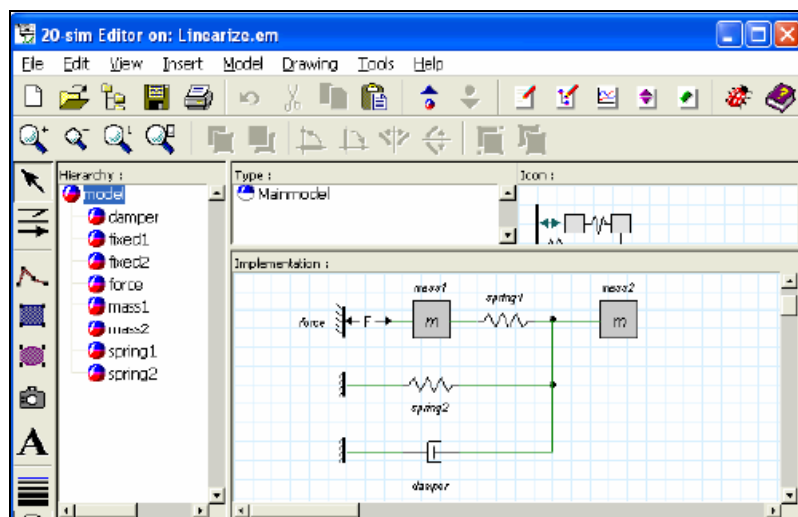


Fig. 7d A modelling example using icons within 20-SIM

4. An example of modelling/ virtual instrumentation

The physical pendulum from the Figure 8 is considered in order to be modelled by a material moving point of mass “m” suspended in a Cartesian coordinate system (Oxy) and loaded by the gravitation. The material point cannot have a free motion; it has to stay on a curved trajectory. The material point performs the motion under the load, but only in accordance to the existing joints [1].

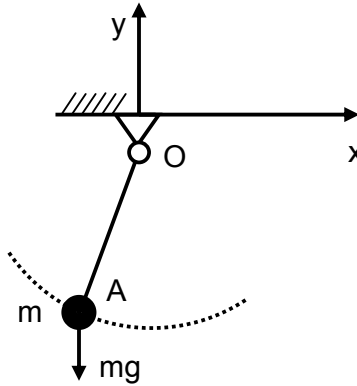


Fig... 8 The physical pendulum

The equation of the trajectory covered by the mass “m” corresponds to a circle with the centre over the point O; this equation is a constraint for the analysed system:

$$x^2 + y^2 - l^2 = 0 \quad (1)$$

The kinetic- and the potential energy of the mass “m” performing the motion are:

$$E_c = m \cdot \frac{\dot{x}^2 + \dot{y}^2}{2} \quad (2)$$

$$E_p = mgy \quad (3)$$

Based on the previous relationships, one can write the Lagrange function characterized by the parameter λ :

$$L = E_c - E_p - \lambda \cdot (x^2 + y^2 - l^2) \quad (4)$$

The equations systems that describe the motion of the material point are:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_k} \right) - \left(\frac{\partial L}{\partial q_k} \right) = 0 \quad (5)$$

for $q = x, y, \lambda$:

$$\begin{cases} m\ddot{x} + 2\lambda x = 0 \\ m\ddot{y} + 2\lambda y + mg = 0 \\ x^2 + y^2 - l^2 = 0 \end{cases} \quad (6)$$

The model can be implemented in any of the simulation environments already existing in the Sensors and Actuators Laboratory within the Department of Mechatronics of the “Politehnica” University of Timisoara: Matlab / Simulink, LabView 8.

In order to make a comparison of the theoretical results with the simulation results, one can use the data acquisition and virtual instrumentation. The accomplished stand that

was used in data acquisition is shown in Figure 9.

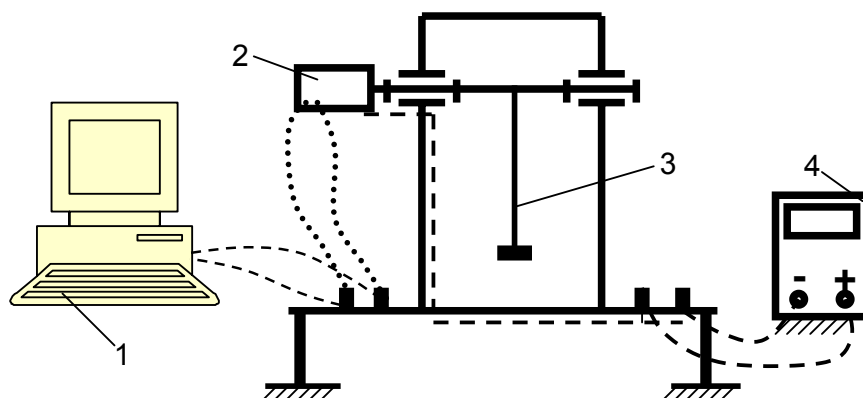


Fig. 9 The experimental stand used in comparative simulation studies (1 – PC; 2 – displacement resistive transducer; 3 – physical pendulum; 4 – c.c. power supply).

Figure 10 illustrates the front panel of the virtual instrument and the acquired image.

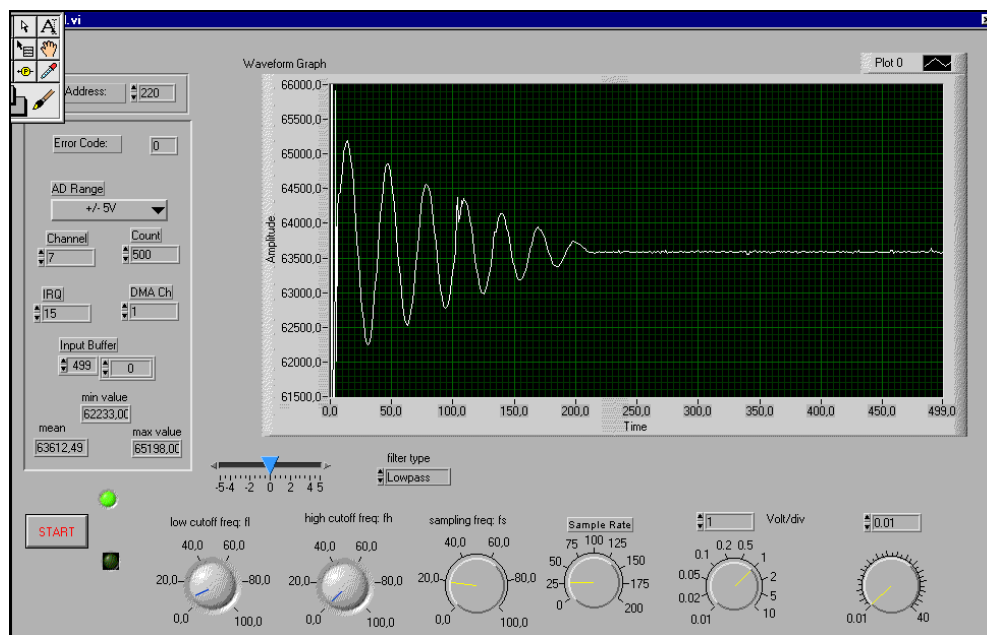


Fig. 10 The acquired signal

5. Conclusions

The authors analysed the various possibilities to model and simulate industrial systems and the reasons that confer a great role to the modelling and simulation processes in the early stages of the lifecycle for an industrial product.

The authors consider the LabView environment as the most appropriate for educational goals, due to the link with the PC.

For complex problems, referring too to the creation of the experiment, Matlab/ Simulink/ dSPACE are recommended.

20-SIM is best for the mechatronic design.

The use of one or another of these environments is the ultimate decision of the user.

Bibliography

- [1] Dolga, V., Proiectarea sistemelor mecatronice, Editura Politehnicii (in curs de aparitie), Timisoara, 2007
- [2] Savii, G.G., Luchin, M., – Modelare și simulare, Editura Eurostampa, Timișoara, 2000
- [3] Necsulescu, D., Mechatronics, Prentice Hall,, New Jersey, 2002
- [4] Bolton, W., Mechatronics, Pearson Education Limited, Prentince Hall, 2003
- [5] Buur, J., A Theoretical Approach to MECHATRONICS DESIGN, Institute for Engineering Design (Tech. University of Denmark), 1990
- [6] Isermann, R., Mechatronische Systeme, Springer Verlag, Berlin, 1999
- [7] Shetty, D., Kolk, A.R., Mechatronics System Design, Pws Publishing Company, Boston, 2000
- [8] ***, LabVIEW Data Acquisition Basics Manual, National Instruments, 1998
- [9] ***, 20-sim, V.3.4, ControlLab Products B.V, Enschede, Netherlands, Help files
- [10] ***, Dymola v.5.1b, Dynasim AB, Sweden, Help files
- [11] ***, Matlab / Simulink, User's Guide
- [12] ***, Modelica Libraires, www.modelica.org/libraries