Creating Dynamic Mechatronics Design with Open Tools

Gordana Janevska and Mitko Kostov
Faculty of Technical Sciences at St. Kliment Ohridski University
Bitola, Makedonska Falanga 33
Bitola 7000, Macedonia
gordana.janevska@uklo.edu.mk

ABSTRACT: *In the current research, we have designed mechatronic models using open tools. We have given importance to the open tool Scilab/Xcos. With the help of simple as well as elegant models, we have presented how to create a dynamic system which contains the physical and mathematical features.*

Keywords: Mechatronics, Model-based Design, Open Source Software, Scilab/ Xcos

Received: 7 January 2022, Revised 15 April 2022, Accepted 10 May 2022

DOI: 10.6025/jet/2022/13/3/68-75

1. Introduction

The design of mechatronic systems could be done with the so-called V model (VDI guideline 2206). A major role in the VDI 2206 guideline has modelling and model analysis in the domains of mechanical engineering, electrical engineering and information technology. Usually, commercial software tools are used to design mechatronic systems. MATLAB/Simulink and its toolboxes are well-known as leading software for model-based design of mechatronic systems, but many others can be added. On the other hand, it is of interest to know Open Source alternatives for it.

A dozen years ago, free software was approved as useful in the area of information technology. The expression “Free and Open Source Software” (FOSS) was introduced to highlight the difference of "free" in terms of free of charge and free access to the source code of the software. Since then FOSS has been developed to a successful business model for computer programs. In information technology there is no doubt in Linux as a very good operating system for network server. The largest supercomputer runs under Linux, and Linux is already a market leader as an embedded operating system for embedded systems. But FOSS is, of course, more than Linux, it also includes other very successful server applications.

From the view point of using software in the field of mechatronics, a small enterprise that develops mechatronic systems according to instructions of the VDI 2206 guideline, should invest in appropriate software and must pay annually high price for software maintenance. Otherwise, it will lose the opportunity for further development of the software, and therefore its custom-

this aspect, the facts about using FOSS as an addition to many useful existing software tools can be considered.

Bruce Perens began using the term FOSS and provides a good analysis of the economic benefits. It includes free access to source code, encouraging other companies, even competitors, to join in software development and share costs so as to accelerate this development.

2. Open Source Alternatives to Matlab/Simulink

MATLAB/Simulink is one of the leading software packages for model-based design of mechatronic systems and market leader in the development and simulation of control systems. This also can be concluded from the MathWorks pricelist. Small engineering offices have financial problems with buying and maintaining the products of commercial software packages. It is therefore good to know some alternatives.

In addition to MATLAB, there are other partially specialized tools such as Modelica/Dymola, ADAMS, ANSYS, ASCET MD/RP, AMESim, CarMaker, DSpace, SimulationX and many others.

Wikipedia lists some alternatives to MATLAB/Simulink at http://en.wikipedia.org/wiki/MATLAB#Alternatives. It is advisable to work with GNU Octave.org and Scilab.org.

GNU Octave, or short Octave, has been developing as a MATLAB clone for decades. It runs on operating systems like MS Windows, Linux, Mac OS X and others. It uses MATLAB files, i.e. MATLAB's Control System Toolbox functions, but a disadvantage is that in Octave there is no graphic editor for building block diagrams such as Simulink in MATLAB. This software can be used, for example, as an introduction to MATLAB, instead of MATLAB, and each student can use it without any problems with licenses. Considering industrial use, there are no interfaces for other programs and extensions for real-time work. Documentation is available on internet, and there already exists some literature.

Another way of modelling systems is proposed by the University of Berkeley with the development of Ptolemy II. This is a promising tool and already contains tools to work in real time. Perhaps it will allow not only modelling, but also realization of experiments in real time, and it would be particularly useful with compilers for rapid software prototyping directly from the tested models.

There are some other free software tools, such as FreeMat which is a MATLAB clone and processes MATLAB scripts, then JMathLib and so on.

Another widely used free open-source software, which is emphasized in this paper is Scilab/Xcos. Scilab/Xcos is a software tool with graphical editor for block diagrams that is comparable to MATLAB/Simulink.

3. SCILAB/XCOS

Developed since 1990 by researchers from the French Government's INRIA ("Institut Nationale de Recherche en Informatique et en Automatique", i.e. National Institute for Research in Informatics and Automation), SciLab is now maintained and developed by the Scilab Consortium after the Consortium was founded in May 2003. It is distributed freely over the Internet and has been open-source since 1994. SciLab is currently widely used in educational and industrial environments around the world.

A key feature of the SciLab syntax is its ability to work with matrices: basic matrix calculations like as transpose are immediately executed, as well as basic operations such as addition or multiplication of matrices. SciLab can also work with more complex objects than numerical matrices. For example, those working in the field of control systems can work with rational or polynomial transfer matrices i.e. SciLab provides a natural symbolic display of complicated mathematical objects such as transfer function, linear systems or graphs.

Starting with version 5.4, which was released in October 2012, Scilab comes along with a convenient user interface, with a files browser, a variables browser, command history, and a command window. The SciNotes editor opens as a separate window.

Similar to Simulink, it has a graphical interface for block diagrams called Xcos, and components library. Xcos is a graphic based

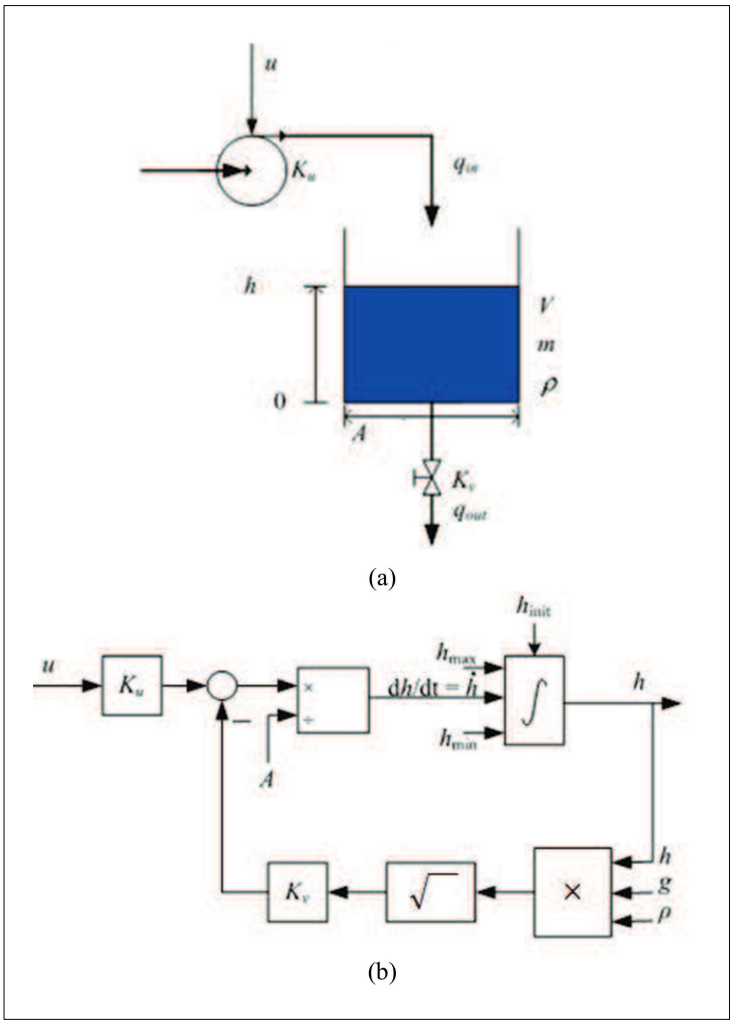


Figure 1. a) Physical model of liquid tank b) Mathematical block diagram of the model

software for modelling dynamic systems. Xcos can create block diagrams that will be used to model and simulate the dynamics of hybrid dynamic systems and models can be compiled in executable code. The new extensions allow the execution of control in real time as well as componentbased modelling of electrical and hydraulic circuits.

4. Case Study

Real system can be presented with a theoretical model through four steps. The real system is simplified, resulting in a physical model and mathematical equations can be written, resulting in a mathematical model. The mathematical model can be solved with a computational model, and the results are visualized by simulation. Xcos is used for the two last steps of theoretical modelling.

4.1. Mathematical Model

The simulated system is a liquid tank, with a pump inflow and outflow valve (Fig. 1). The simulator will calculate and display the level *h* over time. The simulation is in real time, thus giving a feeling of a "real" system. In fact, since the reservoir is a bit slow, the simulation is accelerated in order the simulation time to run faster than real time, to avoid unnecessary waste of time. The user can adjust the input by adjusting the pump control signal, *u*.

Each simulator is based on a mathematical model of the system that needs to be simulated. Accordingly, a mathematical model of the tank should be developed.

By introducing relevant assumptions, a real system can be simplified and afterwards a corresponding physical model can be created. Following assumptions are introduced (parameters used in the expressions below are defined in Figure 1a):

- The liquid is incompressible, i.e. the density of the liquid is ρ = const;
- The tank has straight vertical walls, i.e. A = const;
- The mass and level of the liquid in the reservoir are related to the equation:

$$m(t) = \rho A h(t)$$

- The inlet volume flow through the pump is proportional to the pump control signal:

$$q_{in} = K_u u(t)$$

- The outlet volume flow through the valve is proportional to the square root of the pressure drop of the valve. This drop in pressure is assumed to be equal to the hydrostatic pressure at the bottom of the tank:

$$q_{out}(t) = K_v \sqrt{pgh(t)}$$

Based on the law of conservation of mass, the continuity equation for the liquid in the reservoir can be written:

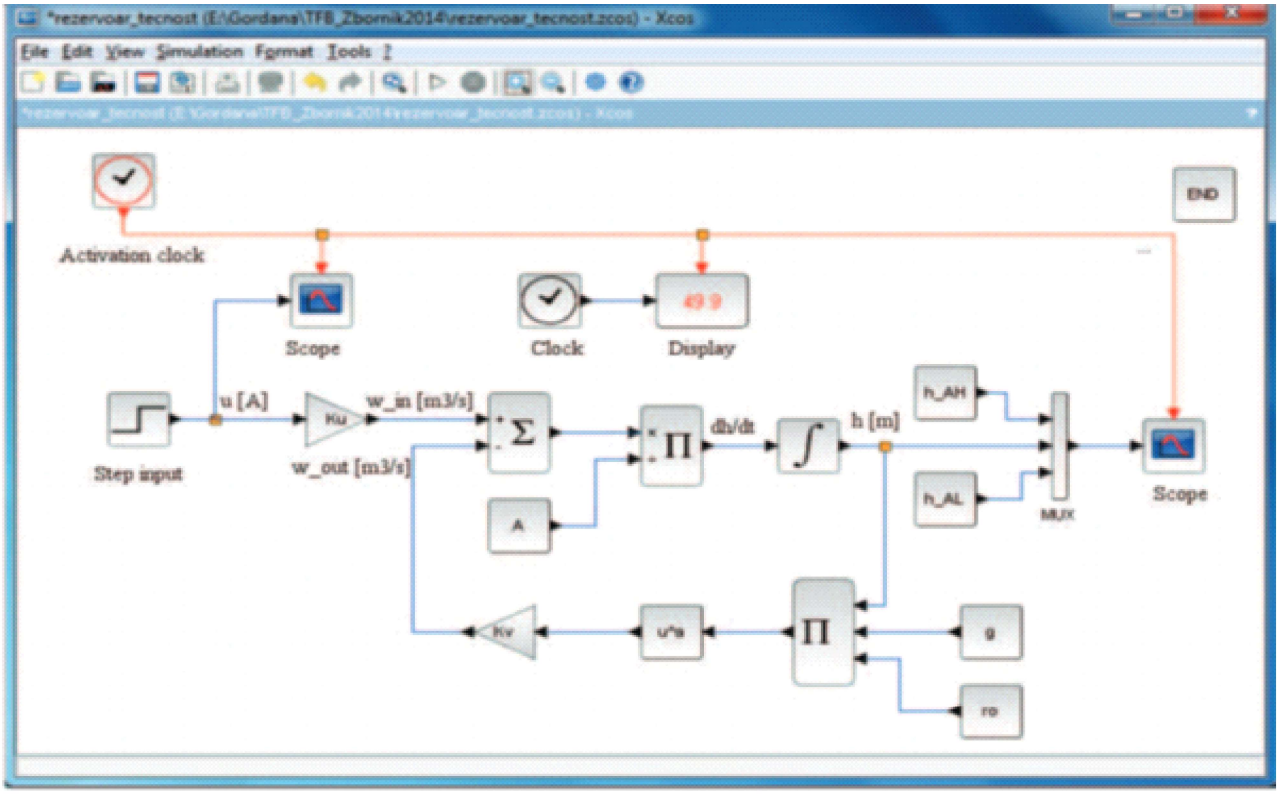


Figure 2. Simulation block diagram of the liquid tank in Xcos

$$\frac{dm(t)}{dt} = \rho q_{\text{in}}(t) - \rho q_{\text{out}}(t) \quad (1)$$

or, taking into account the above relations, it is obtained:

$$\frac{d[\rho Ah(t)]}{dt} = \rho K_u u(t) - \rho K_v \sqrt{\rho g h(t)} \quad (2)$$

Based on the above equation, a mathematical block diagram of the model can be drawn. This block diagram can be implemented in the simulation block diagram. As an appropriate starting point for drawing the mathematical block diagram, the differential equation (2) will be written as a state space model, that is, the first-order time derivative to be alone on the left side, so the differential equation obtains the following form:

$$\frac{d[h(t)]}{dt} = \frac{1}{A}\left[K_u u(t) - K_v \sqrt{\rho g h(t)}\right] \quad (3)$$

The upper equation is a differential equation for $h(t)$, i.e. mathematical model of the considered reservoir. According to it, $h(t)$ can be calculated by using the simulator when $dh(t)/dt$ is integrated with respect to time from 0 to t, with the initial value $h(0)$, which is marked here as h_{init}. Drawing a block diagram of the model (3) can began with adding an integrator to an empty block diagram. The input of this integrator is $dh(t)/dt$, and the output is $h(t)$. Afterwards, blocks with mathematical functions are added in order to construct the expression for $dh(t)/dt$ represented by the right side of the equation (3). The final block diagram for the model (3) is shown in Figure 1b).

4.2. Xcos Block Diagram and System Simulation

Xcos provides many elementary blocks organized in different palettes (Palettes) that can be accessed through the Palette browser - Xcos. Blocks from palettes can be copied to the main Xcos window by clicking first on the desired block, and then at the location where the block should be copied to the Xcos window. The blocks are then interconnected with connections, which is accomplished in a natural way by using the computer mouse. The blocks can be configured by double clicking on the block itself and entering the corresponding numerical or (preferably) context variables (parameters) with a specific name.

The simulation block diagram of the liquid tank built in Xcos is shown in Figure 2.

In the simulation of the level of the considered reservoir with liquid, the initial value of the level is 0.5 m. The control signal of the pump u is 0 till 20 s simulation time, and at 20 s it changes with step 0-0.01A. The simulation is from the starting time 0 s to the final time 50 s. The numerical values of the parameters are:

$\rho = 1000$ kg/m^3 $A = 1$ m^2

$g = 9.81$ m/s^2 $h_{init} = 0.5$ m

$K_v = 0.0005$ $h_{max} = 1$ m

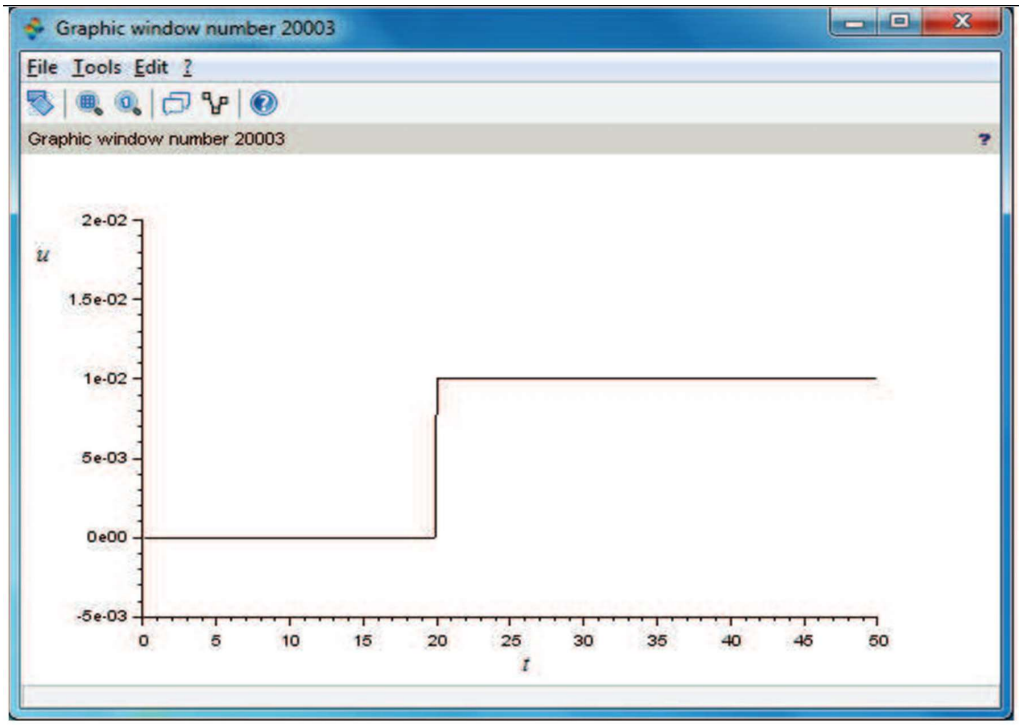
$K_u = 5$ m^3/A $h_{min} = 0$ m

It is assumed that there are level "alarm" limits, which should be drawn along with the level in the simulator. The boundaries are: $h_{AH} = 0.9$ m (alarm high) and $h_{AL} = 0.1$ m (alarm low). The results of the simulation are shown in Figures 3a and 3b.

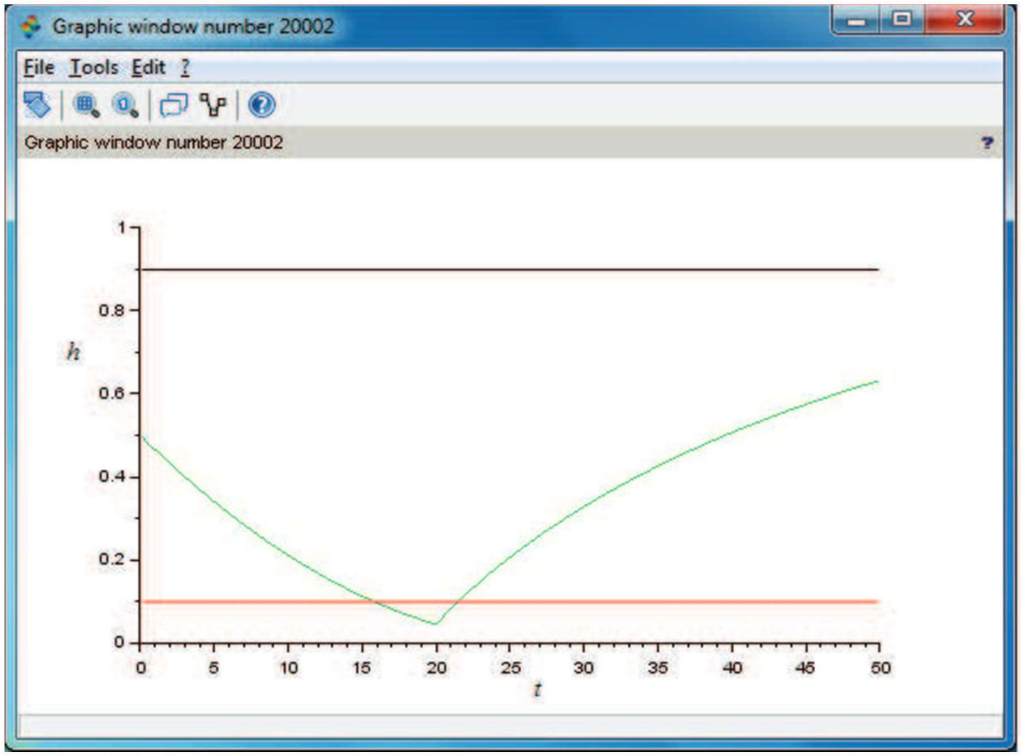
5. Conclusion

In order to enable better organized development of FOSS (Free and Open Source Software) with companies as clients of these tools, OSADL (Open Source Automation Development Laboratory) was established in 2006. Here companies in mechanical engineering, electrical engineering and electronics, computer technology, and universities have been joined in order to define new and open standards in automation.

MATLAB/Simulink alternatives are indicated in this paper. The open source software packages Scilab/Xcos and GNU Octave



(a)



(b)

Figure 3. a) Simulation of the control signal of the pump u; b) Level h and level alarm values h_{AL} = 0.1 m and h_{AH} = 0.9m

are two alternatives that are ready for use in education and industry. The tools are supported by the company (Scilab / Xcos), or by the Internet community (Octave). Documentation, instructions and other help are available online.

The paper also presents some aspects of Xcos formalism. Dynamic systems modelling is presented in a precise and simple manner using the Xcos environment as a model for calculation and simulation, which are the last steps in theoretical modelling. However, it should be noted that the main advantages of MATLAB/Simulink are, for example, the vast amount of pallet tools (toolboxes) and professional support by the MathWorks worldwide.

References

[1] Lohöfener, M. (2008) Design of mechatronic systems and benefit of open source software tools. Italianist, *9th International Workshop on Research and Education in Mechatronics*, Bergamo.

[2] Campbell, S.L., Chancelier, J. & Nikoukhah, R. (2010). *Modeling and Simulation in Scilab/Scicos with ScicosLab*, 2nd edn, Volume 4. Springer: Berlin.

[3] Scilab Enterprises (2013). Xcos for Very Beginners. Scilab Enterprises.

[4] Scilab Open source software for numerical computation, Documentation. www.scilab.org/resources/documentation.