Design of Mechatronic Systems and Benefit of Open Source Software Tools

Manfred Lohöfener[1]

[1]*University of Applied Sciences Merseburg, Faculty of Engineering and Natural Sciences, Merseburg, Germany*

Abstract: The design of mechatronic systems could be done with the so-called V model (VDI guideline 2206[1]). A major role in this guideline has modelling and model analysis in the domains mechanical engineering, electrical engineering and information technology.

Usually we use commercial software tools to design mechatronic systems. A well-known and favorite tool is MATLAB/Simulink, many others should be added. Some results of a questionnaire (VDI/VDE GMA Technical Committee Mechatronics) concerning the use of such tools and features to be developed are discussed. Future tasks are defined.

Open source software has more advantages to be considered than saving license fees[2]. Some examples of open software tools are presented, e.g. Scilab/Scicos or Octave. Requirements concerning characteristics of tools are listed and commercial tools are confronted with open source tools. There are difficulties to use open source tools in the usual business. Questions of warranty, availability and further development are difficult to put into terms of contract.

The founding of the OSADL Open source Automation Development Lab has brought the use of open source software to a new step. OSADL can push the development of open source tools to a commercial usable level. Citation: "The goal of the Open Source Automation Development Lab (OSADL) is to promote and support the usage of open-source software in the context of machine and plant control systems. In principle, it aims to support these industries in a similar manner the Linux Foundation supports industries that provide carrier grade, data center, desktop and mobile systems."[3] We will see how this organization can help us with the development of tools also to design mechatronic systems. On the other hand it has to be determined how universities and companies can contribute to such projects.

Key words: REM 2008, Mechatronics, Open Source, Tools.

I Introduction

Fifteen years ago the expression "Free Software" was associated with hackers or even crackers, i. e. with people developing software in their spare time and distributing it for free. At the same time companies bought the necessary computer programs from traders, or to be more exact they bought the licenses to use the programs. Free software didn't fit in this usual business model of selling and buying, rights and commitments of vendors and customers.

Ten years ago free software was approved to be very useful in the area of information technology, and the expression "Free and Open Source Software" FOSS was introduced to clear up the difference to "free" in the sense of free beer. Instead of this meaning FOSS stands for free access to the source code of the software. Since then FOSS has been developed to a succesfull business model for computer programs. In information technology there is no doubt in Linux as a very good operating system for network server. The most super computer work under Linux, and on the other hand Linux is already the market leader as an embedded operating system on embedded systems. But FOSS is of course more than Linux, it comprises also very succesfull server applications like Apache webserver, email server etc.

But there is a delay in using FOSS applications on computer desktop, although we can work with such software for nearly all daily tasks in our office. The autor himself tests the work with FOSS programs successfully in self-experiment since more than seven years.

Now to another area of software use, to mechatronics. Let us assume a small enterprise developing mechatronic systems according to instructions of the guideline VDI 2206, the guideline to develop such systems using model based design. This company has to invest in suitable software and has to pay annually high prices for software maintenance. Otherwise it would loss the contact to further development of software and consequently to its customers! So we have to ask, if we could reduce costs using FOSS without loosing the benefit or even with an outstanding in-

crease of the benefit. It sounds to open a new competition, maybe a displacement of commercial software. No, we don't need the software rebels fighting against capitalistic software companies, but we will have a look at the facts of FOSS as an addition to the very useful existing software tools.

Bruce Perens launched the expression FOSS and gave a good analysis of the economic benefit. It includes the free access to source code, the encouraging of other companies, also even competitors, to join in the development of software and to share its costs and so to accelerate this development. Of course, nobody has to open his company know-how. But analyses disclosed, that usually only about 5 % of the software used in a company containes such specific know-how, but about 95 % of the software could be developed in a community to the benefit of all participants because this part don't contribute to the market position, to the sales of the single company.

II Software in Mechatronics

But what software tools do we use in mechatronics? This was an important question of a questionnaire initiated by the Technical University of Dresden. Unfortunately this questionnaire couldn't been evaluated because of the low number of answers. However we can get a spectrum of programs for mechatronics and some statements.

As expected the most used tool for the design of mechatronic systems is MATLAB/Simulink. This and other tools are mainly used in control design, mechanical design, test and verification. Analyses are done mostly in the range of milliseconds, and the most used description form is ODE (ordinary differential equation). Beside MATLAB there are called other partly specialized tools like Modelica/Dymola, ADAMS, ANSYS, ASCET MD/RP, AMESim, CarMaker, dSpace, SimulationX and much more.

But there is no FOSS tool listed in the questionnaire, although some programs are available. So we will introduce some of these tools.

III FOSS in Mechatronics

A. Octave

A rather old program is Octave[4]. It defines itself to be a MATLAB clone, and it is steadily further developed. It uses MATLAB script files, comprises actually functions of the so-called Control System Toolbox (MATLAB), but has no graphic editor like Simulink.

GNU Octave, version 3.0.1
Copyright (C) 2008 John W. Eaton and others.

So we can print a step answer of a 2nd order transfer function (example) using only one line:

```
octave:1> step(tf([1 5],[1 1 1]));
```

The result is given in Figure 1.

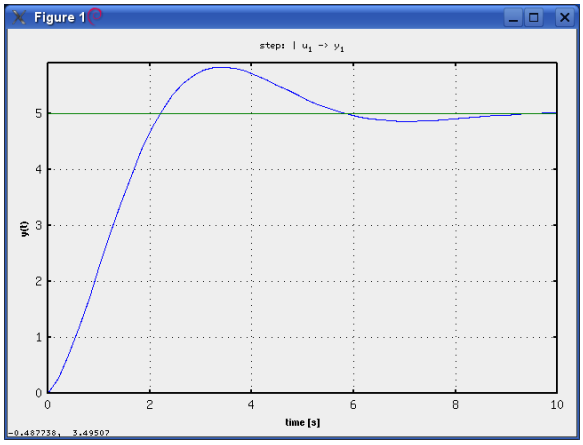

Figure 1: Plot of Octave

We can use this program for instance in an introduction to MATLAB instead of MATLAB, and every student can use it without having problems with licenses. For an industrial use we miss interfaces to other programs and real time extensions. Documentation is available online, and an interesting book is [5].

B. Scilab

Scilab/Scicos[6] is a software tool with a graphic editor for blockdiagrams comparable to MATLAB/Simulink.

scilab-4.1.2
Copyright (c) 1989-2007
Consortium Scilab (INRIA, ENPC)

An example is given in Figure 2.

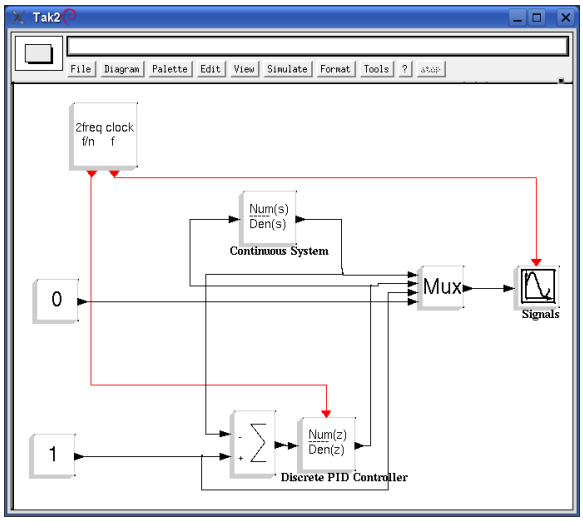

Figure 2: Discrete PID Control

The resulting signals are shown in Figure 3.

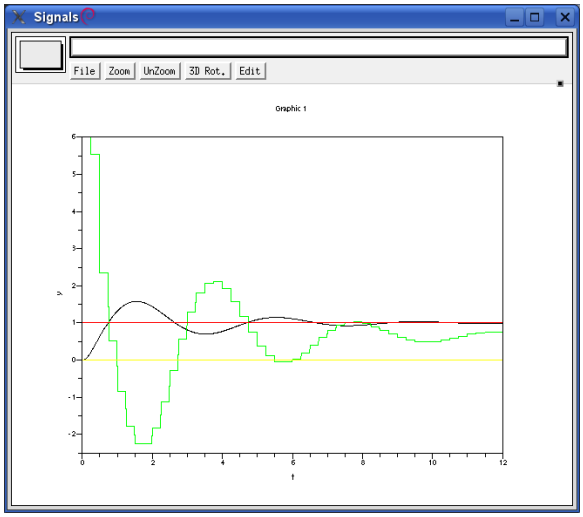

Figure 3: Signals

But Scilab/Scicos comprises also domain specific modelling using Modelica. There are palettes with blocks for electrical and thermohydraulics, an example is shown in Figure 4.

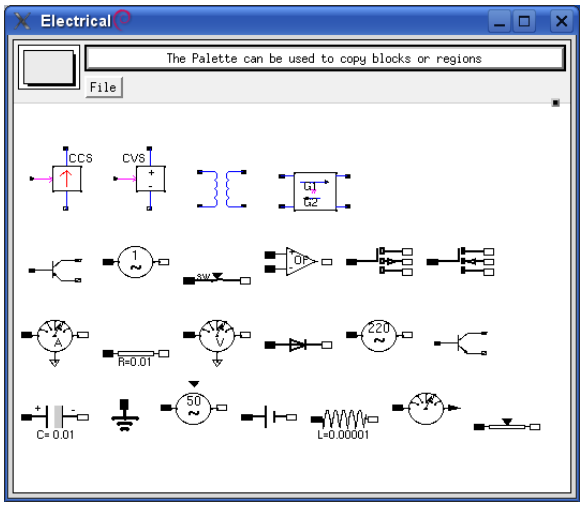

Figure 4: Palette with electrical blocks

For our use in mechatronics it is of course necessary to have an interface to the real world, for simulations in a real environment. Therefore a number of tools were developed and can be used with the real-time operating system RTAI Linux. So we have additional palettes with interfaces to real processes. There is the input output framework COMEDI and the real-time environment RTAI-Lab. But in the last few years Linux was further developed to have genuine real-time characteristics, so we have to translate all the succesfull work for RTAI to the standard Linux kernel with real-time patches, and this is a promising way. We can assume to get a very powerful tool, if we also manage the step of automatic code compilation for several target systems, and some steps are already done. Detailed informations can be found in [7].

C. *Ptolemy II*

Another way of modelling systems is shown from the University of Berkeley with the development of Ptolemy II[8]. This is a promising tool and contains already real-time work as shown in Figure 5 in an example delivered with Ptolemy II.

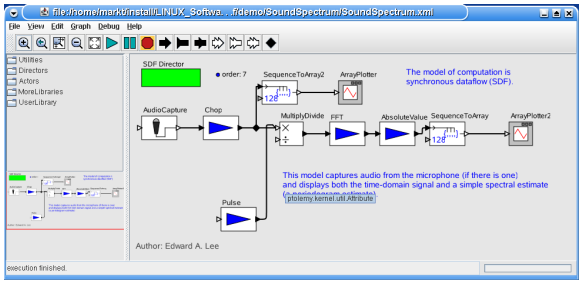

Figure 5: Synchronous data flow

Here a signal from the sound card input is processed with Fast Fourier Transformation FFT, and we get a plot with the actual data like in Figure 6.

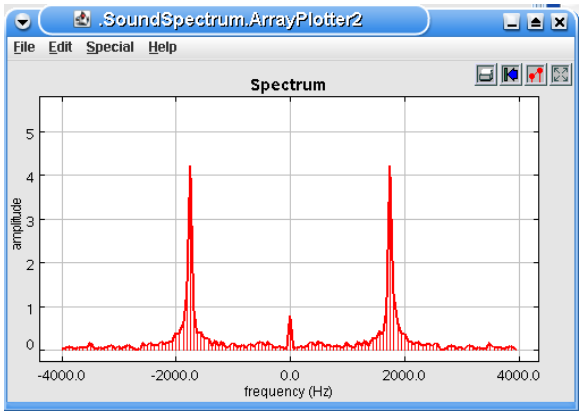

Figure 6: FFT spectrum

Maybe it is a way not only for modelling but also for real-time experiments. It would be very useful with compilers for rapid software prototyping directly from the tested model.

D. *Other tools*

There are some other FOSS tools to be numbered shortly without being tested by the author.

A MATLAB clone is called FreeMat[9] and processes MATLAB scripts.

JMathLib[10] is a further tool and presents itself to be: “A Java clone of Octave, SciLab, FreeMat and MATLAB”. Maybe JMathLib will become a very useful tool.

IV Future Tasks

Obviously all presented programs have lacks with inhibit the use as productive tools in research and development. A way out of this dilemma could be a bet-

ter organized development of FOSS with companies as customers of these tools. One organization to meet such needs was founded in 2006, the Open Source Automation Development Lab OSADL. Here companies from mechanical engineering and machine tools, from electrics and electronics, from computer technology and also universities are joined to define new and open standards in automation.

A first important step was the definition of a widely scalable real-time operating system for all processors and systems, and the standard Linux kernel with real-time patches was chosen. The usability has been proven in numerous applications. Next steps could handle PLC programming etc.

The author would propose to add the development of software for design of mechatronic systems to the tasks of OSADL. Of course it would be useful, if interested enterprises and universities join OSADL and support and promote such tasks.

References

1] VDI, Verein Deutscher Ingenieure (Ed.). Design Methodology for Mechatronic Systems. VDI guideline 2206. Berlin: Beuth 2004

2] M. Lohöfener. Experience with the Use of Linux an a Desktop PC and in Embedded Systems. 5th International Workshop on Research and Education in Mechatronics REM 2004, 1st – 2nd October 2004, Silesian University of Technology, Kielce-Cedzyna, Poland

3] OSADL Open source Automation Development Lab. http://www.osadl.org/

4] Octave Home. http://www.gnu.org/software/octave/

5] Quarteroni, Alfio; Saleri, Fausto: Scientific Computing with MATLAB and Octave. Berlin: Springer 2008

6] Scilab Home. http://www.scilab.org/

7] Campbell, Stephen L.; Chancelier, Jean-Philippe; Nikoukhah, Ramine: Modeling and Simulation in Scilab/Scicos. Berlin; Springer 2005

8] Ptolemy II Home. http://ptolemy.berkeley.edu/ptolemyII/

9] FreeMat Home. http://freemat.sourceforge.net/

10] JMathLib Home. http://www.jmathlib.de/