Computer Support for Mechatronic Control System Design

Job van Amerongen, Theo J.A. de Vries and Erik Coelingh

Drebbel Institute for Systems Engineering,
Control Laboratory, EE. Dept, University of Twente, EL–RT,
P.O. Box 217, 7500 AE Enschede, The Netherlands.
E-mail: J.vanAmerongen@el.utwente.nl, http://www.rt.el.utwente.nl/amn

**This paper was presented in the invited session on
'Computer Support for Mechatronics System Design'
at the International Conference on Advanced Mechatronics
Okayama, Japan, 3-5 August 1998**

Computer Support for Mechatronic Control System Design

Job van Amerongen, Theo J.A. de Vries and Erik Coelingh

Drebbel Institute for Systems Engineering,
Control Laboratory, EE. Dept, University of Twente, EL–RT, P.O. Box 217, 7500 AE Enschede, The Netherlands.
E-mail: J.vanAmerongen@el.utwente.nl, http://www.rt.el.utwente.nl/~amn

Keywords: mechatronics, CACSD, controller design, modeling, simulation

Abstract: This paper discusses the demands for proper computer aided control system design tools for mechatronic systems and identifies a number of tasks in this design process. It puts the contributions to the session in perspective and describes how the papers in this session (partly) contribute to a solution. Some issues not addressed in the session are discussed and the current state-of-the-art as perceived by us is described. Real mechatronic design, involving input from specialists from varying disciplines requires that the system can be represented in multiple views, including a view of the mechanical construction, time domain behavior, frequency characteristics etceteras. Several tools are already available but there are still substantial shortcomings. The paper gives indications about the developments needed to come to better design tools in the future.

1. Introduction

A useful definition of *mechatronics* is: "a technology which combines mechanics with electronics and information technology to form both functional interaction and spatial integration in components, modules, products and systems" (Buur, 1990). This technology differs from classical design patterns, where the design starts with the mechanical subsystems, then the electrical subsystems and finally the controllers. In order to form functional interaction and spatial integration, subsystem designs need to overlap, and hence simultaneous involvement of several disciplines needs to be realized in a coordinated way. The design process of controlled electro-mechanical systems with a mechatronic design approach is therefore generally more complex, but leads to systems with a superior price-performance ratio.

As Simon (1973) states, a design problem in general is "ill-structured in the large, but well-structured in the small". That is, the complex, difficult to tackle design problem can be mastered by splitting it up into small well-structured problems for which a 'local' solution can be found (Figure 1).

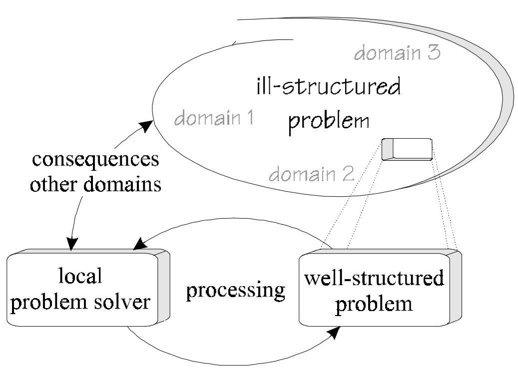

Figure 1 Schematic structure of solving ill-structured problems (Adapted from Simon (1973))

In the same way, the complexity of mechatronic design can be tackled. However, the solutions to the well-structured smaller problems should not —as in the classical approach— be considered just as 'local' solutions. Solutions to domain specific sub-problems should be formulated while taking into account the consequences of the solution in other domains and by considering alternative solutions in these other domains.

When designing systems with this approach, the use of computer tools is indispensable. Such tools facilitate (automated) manipulations of the proposed design, allow to record and browse through relevant knowledge and experience, and document the process. The rapid technological developments and the continuing need for reduced times-to-market of new products demands more and more advanced tools in the design stage.

In this paper and session, we focus on aspects related to Computer Aided *Control System* Design (CACSD) of Mechatronic Systems. We present and discuss the latest developments in this area. In section 2, we try to put the contributions to the session in perspective. After that, we discuss the issues not addressed in the session and give a brief indication of the current state-of-the-art as perceived by us. We end with conclusions.

2. Example

When a new car is being designed several aspects are important:

- look of the vehicle
- aerodynamic properties
- weight of the vehicle
- driving properties
- safety systems, such as air bags and ABS
- electronic controls of engine, AC-installation etc.
- manufacturing aspects as well as maintenance and recycling issues
- price, etceteras

Although some of these design issues seem to be hardly related, they all influence each other. As stated in the introduction, the total problem is ill-structured, but parts can be solved in a well-organized way. If there is a well-defined interface between the different parts of the design, these parts can be tackled more or less separately. For, instance, the weight distribution in the car influences its driving properties that could be investigated by means of simulation tools. It is almost obvious that computer simulations with the right parameters of the vehicle could have predicted, and thus prevented, the problems with the small Mercedes Benz car well beforehand.

To do simulations of the driving properties it is not needed to have a detailed description of all the components involved. But it is essential that the weight distribution is well known. This requires an interface that, for instance, easily enables changes made from an aesthetic point of view to be evaluated in dynamic simulations. On the other hand, the dynamic simulations should give indications about how certain components should be changed or placed in a different location. In addition, where design decisions are more closely related, a proper and more detailed interface is even more important. This demands for a kind of “core language” or “core model” that represents the design object internally in the computer and that can be viewed from various windows. Only then proper information can be given to all designers involved.

The car industry seems to be well aware of the need for advanced modeling and simulation tools. Confronted with the need to react quicker and quicker in a highly competitive business, they have to rely more and more on simulation tools. The Control Laboratory was involved in a European project intended to develop a library structure with well tested, *reusable* models for mechatronic system design. (Olmeco, 1991). One of the companies involved in this project was PSA (Peugeot, Citroen) in France. In this library bond graphs were used to represent the models.

When we would focus on the design of a control system alone, a representation in the form of transfer functions or a state space description would be appropriate for linear or linearized systems. From such a representation plot windows with the behavior in the time domain, frequency domain or s-plane can easily be given. In addition, performance criteria can be displayed, indicating whether the system performs optimally. It would be a great advantage if changes made in one domain (window) were reflected immediately in the other windows. Even such simple systems are not yet commonly available. In order to realize them fast or parallel computers will be required.

3. CACSD in a mechatronic setting

In the light of the characterization of the design process given in Section 1, we can distinguish the following tasks of the mechatronic control system designer:

1. Isolation of a *well-structured control problem* specification from an ill-defined system design specification
2. Formulation of a *competent plant model* given an incomplete and indefinite system design proposal
3. Selection of an *appropriate control strategy*
4. *Implementation and realization* of a control system according to the selected control strategy
5. Assessment of the attainable *performance and expected cost* (in terms of time, effort and money) of the controlled system
6. *Communication* of the consequences, expectations, demands, desires, etc. with respect to the proposed system design from the perspective of the control engineers

Although presented as separate tasks, the above items have a strong mutual dependence. The important implication of this is that design software that addresses just one of these tasks is not very useful. In general terms, we can state that the more tasks are supported, the more powerful a computer tool will become.

4. Existing tools

4.1 *Matlab/Simulink*

In controller design, CAD tools are reasonably well developed and used. The well known and widely used package Matlab (and similar packages, such as Matrix_X) is now a standard tool for the control engineer and is part of many modern text books on control. Matlab supports the implementation and realization of mainly linear control systems (task 4) by using plant models in the form of state space descriptions and transfer functions, in the Laplace domain, frequency and time domain (optimal control). Non-linear plant descriptions can be considered in the design process by means of the Simulink toolbox. Simulink is a simulation program based on block diagram input; it can be used to obtain a linear plant model (task 2) and to assess the performance of the controlled non-linear system (task 5). Many programs provide interfaces to Matlab. Products like dSpace translate controllers designed in Matlab directly into code for a digital signal processor (DSP), enabling fas protoyping.

In this session, the other papers discuss one or more of the above-mentioned tasks and describe tools to handle the related design issues.

4.2 *Schemebuilder Mechatronics*

Counsell and Porter (1998): focus on task 3 (quite unique). Schembuilder Mechatronics also addresses task 2, in terms of *physical system models*, which is a more powerful way than Simulink, and addresses task 4 and task 6.

4.3 *CSDA*

The system of Maekawa (1998) is an extension to Matlab. The system supports tasks 1-5. Quite unique are the modules for task 1 and 3. The system specifically intends to let inexperienced designers complete a satisfactory control system design.

4.4 *MATX/RTMATX*

The system of Koga and Sampei (1998) focuses on task 4, and provides an alternative for the Matlab Real Time Workshop and Simulink. Unique feature is that routines for the design of control systems can be included in the experimental software.

4.5 *20-sim*

Coelingh et al. (1998) address in their paper support for task 5. New in here is that already at an early stage of the design process, when many decisions remain to be taken, a reasonably accurate performance assessment can be made. In related work, support for tasks 1, 2, and 6 has been discussed. Like Counsell and Porter, Coelingh uses *physical system models* in the form of bond graphs as available in the 20-sim environment. The computer program 20-sim (pronounce 'Twente Sim'), developed at the University of Twente, supports input of models in the form of bond graphs, in addition to model input in the form of block diagrams and equations. It supports sub models, organized in a hierarchical way. This makes the program well suited to reuse models developed in earlier projects. Version 3.0 of 20-sim, expected to be launched in the beginning of next year, fully supports an object oriented approach to modeling, in addition to a number of other new features (Breunese, 1996, De Vries, 1994). 20-sim supports the modeling process itself and has an extremely fast simulator, through its build in compiler and advanced simulation algorithms. As an interface to other environments, a C-code generator automatically converts complete models or sub models (e.g. controllers) into C-code in the form of ANSI-C functions or ANSI-C stand alone code, as well as Simulink S-functions.

5. Task 6: Communication

A major future issue will be the support for task 6. First of all, this will involve appropriate training and habit forming of the members of a mechatronic design team. Computer support can help here if it would enable each member of the design team to work with the tools most suitable for him, but still using up to date system models that consistently describe one and the same proposed design. This 'dream situation' is far beyond the current state-of-the-art. However, promising developments are being made, which can be categorized in two groups:

- A number of 'general purpose' modeling systems, both commercial and academic, is becoming available that allows to build dynamic behavior models using discipline-specific icons. Examples are, e.g. Saber (ref. Saber), ICAP (ref ICAP), the before-mentioned 20-sim 3.0, Schemebuilder (ref. Schemebuilder) and Camel (ref Camel). This is important, as functional interaction is generally realized through dynamic interaction; the mentioned software systems hence allow to model this interaction at a system level, while representing discipline-specific parts of the model in discipline-specific ways.
- Domain-specific modeling systems are given the capability to exchange data while doing analyses. For example, the mechanical dynamic modeling environment DADS can be linked to Matlab. In this way it can run simulations of controlled mechanical plants (ref. DADS). Typically, these connections make use of techniques like Object Linking and Embedding (OLE2) and Dynamic Data Exchange.

In our view, two problems yet remain to be attended by the scientific community:

1 Typically, models that are competent for aspects important to one discipline, are incompetent in another discipline. That is, a transformation of the models is needed. For example, a DADS model of a mechanical plant is unwieldy and too complex to be used for control system design; it needs to be simplified and stated in other terms first. The formulation of appropriate transformation algorithms and the (speedy) realization thereof in software is an important issue.

2 When the number of software tools contained in the tool suite (or the abilities of a single tool) grows, there is an increasing need for support in order to guarantee consistent models and to use the tools in a coordinated manner. This requires a proper interface.

6. *Interface*

A well-defined interface is crucial to enable different representations of the design object, while simultaneously maintaining a consistent model. Control engineers are used to models in the form of transfer functions or state space descriptions. Without special precautions, they generally do not have a direct relation to the physical parameters in the system. This is a serious disadvantage when not only the structure and parameters of the controller are being considered. If the physical system that has to be controlled is not taken for granted, suggestions made for modification of the physical system during the design of the controller have to be translated into physical parameters. This implies that the parameters in the transfer function should be directly and dynamically related to the parameters of the physical system. In another paper in this session, an example is given of the design of a plotter (Coelingh, De Vries and Van Amerongen, 1998). The design method described there is a first step towards the goal of a design system that directly relates modifications in the controlled system to changes in the construction of the plotter. The parameters in the simplified model have a known —may be complex— relation to physical parameters of the plotter. In that paper the bond-graph language is used to relate the physical reality to the more simple transfer function based model used for the controller design. Bond graphs are able to capture all the relevant physical characteristics, necessary to do simulations as well as to indicate which physical parameters should be changed in order to achieve the desired performance. The bond-graph representation is domain independent, but domain specific models can easily be converted into a bond graph. In principle, bond graphs can be automatically converted into domain specific models, provided that additional information about the domain as well as geometrical information is stored together with the bond graph (De Vries 1994). The conversion from a model that is closely related to a domain-specific physical system into a more abstract model is relatively easy and is a one to one mapping. But after simplification the more abstract model can not be converted back by a one to one mapping into the original model domain. This introduces ‘additional freedom’: one parameter in the more abstract model maps to various combinations of parameters in the physical model and not all combinations may be feasible. The physical parameters will be subject to constraints, which will limit the real freedom of choice.

7. Multiple views

With a proper interface, each designer can work with the models and kind of output most appropriate for the specific design task. But even for one design task multiple views on the same problem can considerably enhance the insight into the problem and help to find solutions. Consider for example the design of a filter that should suppress the resonance frequencies present in a measured signal. By modeling these frequencies as sinusoidal measurement noise, a Kalman filter can be designed that is well able to suppress these frequency components (Figure 1). A simulation of the complete system demonstrates that (Figure 2). However a view in the frequency domain of the transfer from the measured signal (1) to the observed signal (2) reveals that the Kalman filter is in fact a notch filter (Figure 3). This view shows the width of the frequency band that is being suppressed and could suggest filter modifications in order to achieve suppression of the disturbances over a wider range of frequencies. Of course the ideal situation would be that all views are generated simultaneously, without a noticeable delay and that modifications could be made in any window.

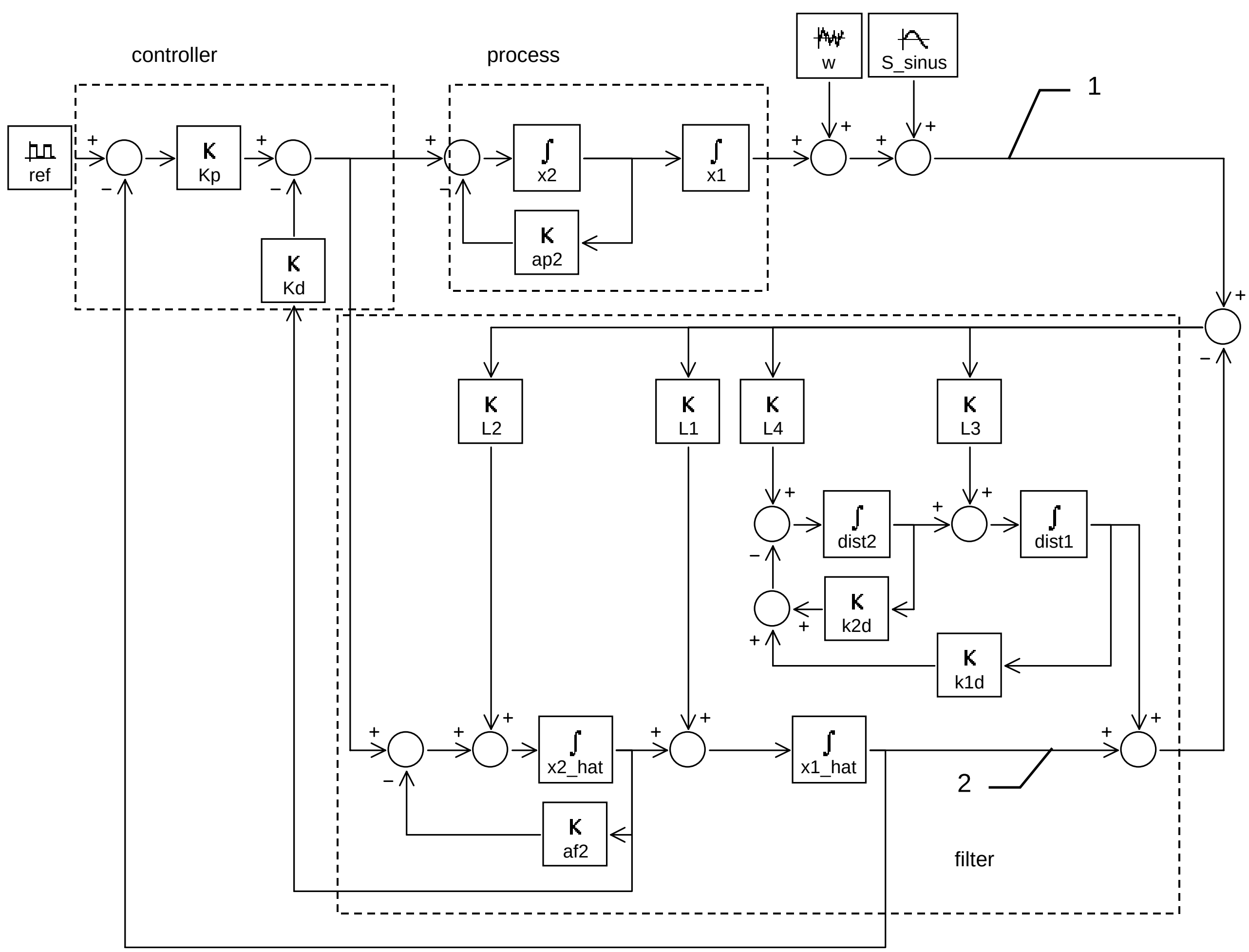

Figure 2 Kalman filter for estimation of non-measurable signal and for suppression of resonances in measuring device

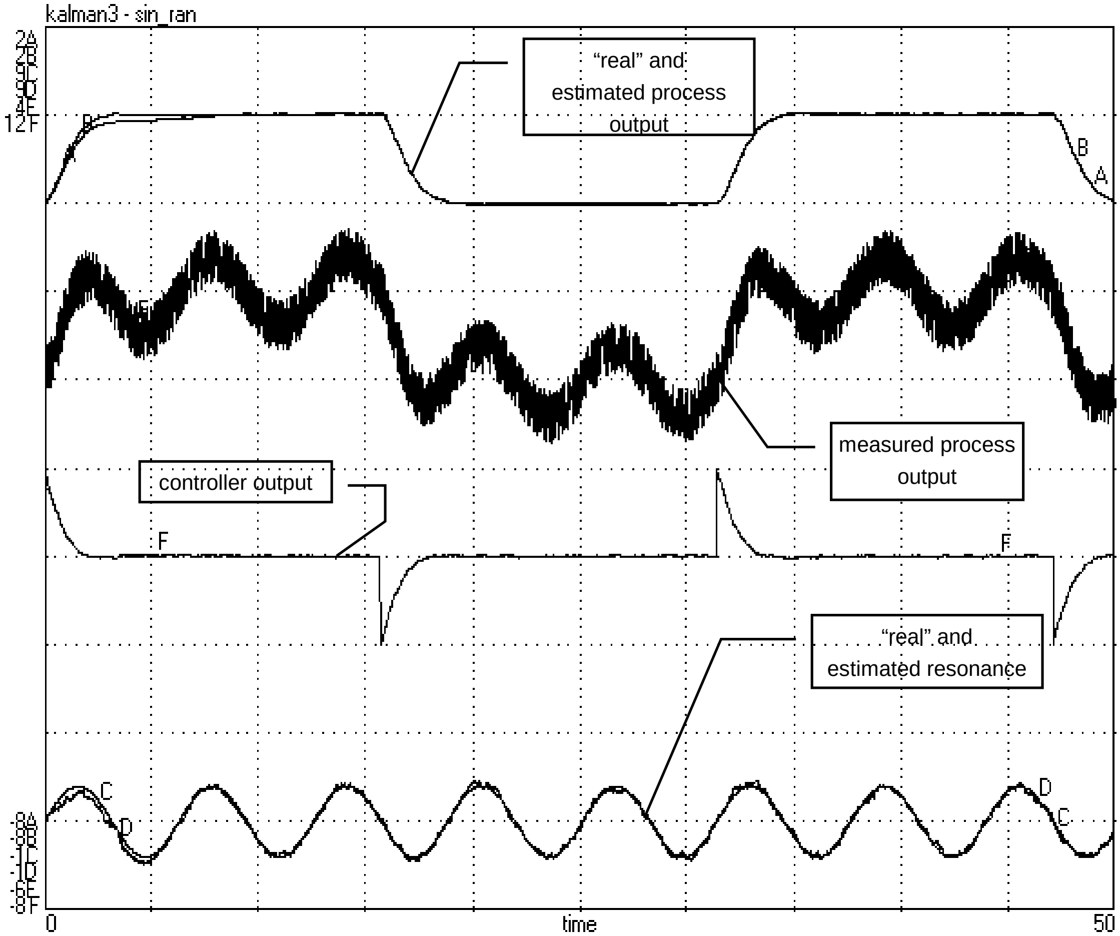

Figure 3 Time responses of the various signals

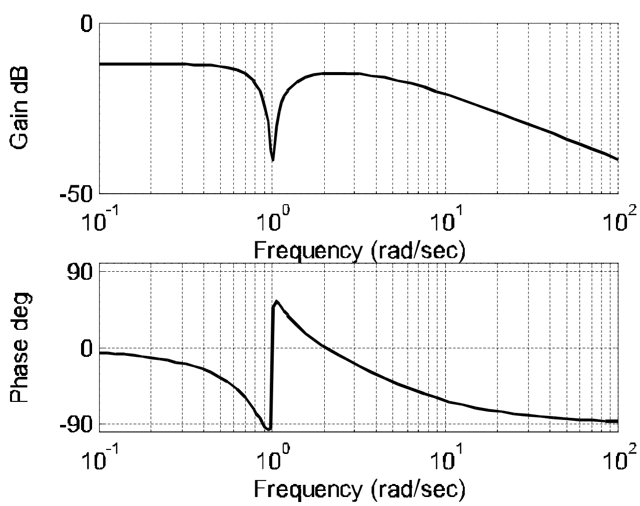

Figure 4 Kalman filter viewed in the frequency domain as a notch filter

7.1 *Virtual reality*

One of the 'views' that could enhance the feeling designers can get of their own and others decisions during the design process is a 'virtual reality view'. In the November 1997 issue of IEEE Spectrum, Kaplan (1997) describes the use of virtual reality tools in the design of new cars. Visual displays, possibly in head mounted displays to generate a 3D-image, and haptic displays give the views and feeling of the virtual reality. These systems not only contribute to a shorter time to market because the virtual representations limit the need of building prototypes. It also enables global digital prototyping. The information can easily be exchanged world wide, enabling a design team with designers working in places scattered around the globe in different time zones, to work 24 hours a day. Standards for the exchange of the graphical information are available.

8. Conclusion

In this paper it has been shown that the design of a mechatronic system can be split into a number of distinct tasks. The other papers in the session describe several of these tasks more into detail and present tools to solve these tasks. A single design environment able to cover all these tasks is not yet available. From the perspective of control system design Matlab is a de facto standard for the design of all the elements of the system that are directly related to the controller. With additional toolboxes such as the Simulink toolbox (for non-linear simulations), State flow (for the design of the computer-based controller), as well as additional hardware such as dSpace (for the realization of controllers in DSP's), prototypes of control systems can relatively easily be realized. But from a mechatronic design perspective the standard Matlab tools are not sufficient. The tools presented in this session are examples. For instance, modeling of mechatronic systems may be much easier with a modeling and simulation program such as 20-sim, that supports modeling in terms of physical components, than with a block diagram based tool like Simulink. The need for a proper interface between the different tools has been identified as one of the key issues to come to a real mechatronic design approach. Multiple views on the design including the possibility to let changes in one view, simultaneously and instantly, influence the views in another domain are essential for a real mechatronic design environment.

8. References

20-sim, http://www.rt.el.utwente.nl/20sim

Breunese, A.P.J., "**Automated Support in Mechatronic Systems Modelling**", Ph.D. thesis University of Twente, 1996

Buur, J. "**A theoretical approach to mechatronics design**, Ph.D. thesis, Institute for Engineering Design, Technical University of Denmark, 1990

Camel: http://MLaP.Uni-Paderborn.de/

Coelingh, Erik, Theo J.A. de Vries and Job van Amerongen, "Automated Performance Assessment of Mechatronic Motion Systems during the Conceptual Design Stage", International Conference on Advanced Mechatronics, Okayama, Japan, 1998

Counsell, John M. and Ian Porter, "Schemebuilder Mechatronics: Design Principles for Controller Design", International Conference on Advanced Mechatronics, Okayama, Japan, 1998

DADS: http://www.cadsi.com/, http://www.cadsi.com/dadsplant.html.

dSpace: http://www.dspace.de

ICAP: http://www.softsim.com/eanal/icap4o.htm

Kaplan, Gadi, "Auto manufacture digitizes in depth", **IEEE Spectrum**, November 1977

Koga, Masanobu and Mitsuji Sampei, "An Integrated Software Environment for Design and Real-Time Control of Mechatronic Systems", International Conference on Advanced Mechatronics, Okayama, Japan, 1998

Maekawa, Kyoshi, "Control System Design Automation for Mechanical System -Machine Tool Module and Control Parameters Calculation Module", International Conference on Advanced Mechatronics, Okayama, Japan, 1998

OLMECO: Open Library for Models of mEchatronic COmponents, Part I-III, Esprit Proposal EC 6521, Brussels, Belgium, 1991

Saber: http://www.analogy.com/, http://www.analogy.com/products/prod.htm#ControlSystems

Schemebuilder: http://www.comp.lancs.ac.uk/edc/schemebuilder.html

Vries, Theo J.A. de, "**Conceptual design of controlled electro-mechanical systems**" Ph.D. thesis University of Twente, 1994