CAMP-G / SysQuake, an Integrated Environment to Understand Dynamic Systems and Design Controllers

Yves Piguet, Ph.D.
Calerga and Swiss Institute of Technology
Lausanne, Switzerland
www.calerga.com
yves.piguet@calerga.com

Jose J. Granda, Ph. D.
Department of Mechanical Engineering
California State University
Sacramento, California 95819, USA grandajj@ecs.csus.edu

Gianni Mocellin, Ph.D.
Straco Consulting
Geneva, Switzerland
straco@swissonline.ch

Keywords: Modeling; Bond Graphs; Computer-Aided Control System Design; software.

ABSTRACT

Understanding a physical system and designing a controller imply two major tasks presented here with an integrated software environment. The first one which occurs even before using a computer, one needs either the transfer function of the controlled "plant", a set of differential equations that describe it, or a block diagram of the system. The second task is the real design and institutive phase. Using a set of physical parameters, one can perform a simulation in the time or in the frequency domain. At this point the question is to satisfy the design requirements. A software package CAMP-G/SysQuake has been developed and integrated to assist in these two important areas. The first task of generating the model, the appropriate differential equations and transfer functions is performed by the bond graph modeling software CAMP-G, with a software interface, the second aspect is accomplished in an intuitive and instantaneous manner, the response in time and frequency domain is performed with SysQuake. Changes to the response due to parameters are instantaneously displayed. Using real time computer graphics. A combination of analysis and synthesis is the result of this development in order to provide a tool, which gives instantaneous insight of what the design would be upon changes of the physical parameters or control gains.

INTRODUCTION

The problems encountered by the simulation and control engineer have not radically changed since the introduction of automatic control. Engineers were facing systems whose performances they wanted to improve. In order to get a good feedback loop, they started to design such loops according to frequency representations (Nyquist, Nichols). During the fifties, state-space representations were introduced, allowing online parameter estimation (Kalmann). Noticing that modern control did not take into account some fundamental design criteria like robustness margins, they stepped back to improved frequency methods like H_∞ or QFT [1], allowing the integration of both methods.

On the other hand, noticing that modern computer systems allowed a radically different approach to modeling graphical Computer-Aided Design tools like Camp-G (Granda 1982) [2] were introduced, allowing a fast model generation (symbolic as well as numerical). Finally, other researchers were not satisfied by the possibilities offered for generation of graphics and lack of interactivity, followed a brand new way which led to SysQuake [3]. Such revolutionary software allows the intuitive comprehension of parameter influence and sensitivity analysis on the response of systems. Some very interesting applications have already demonstrated the power of this new concept: fast prototyping [4], multi-model synthesis of a robust polynomial controller [5], or teaching of automatic control [6].

The two packages complement each other from the modeling phase to the design of a controller forcing the system to have the desired behavior. A link has been developed between these two technologies resulting in an integrated environment. This paper is intended to describe such process and demonstrate its power using examples familiar to those in the field of control systems who use different methods other than bond graph modeling. CAMP-G, the Computer-Aided Modeling Pro-gram with Graphic input, is used to generate the models and then SysQuake, performs instantaneous analysis and design based on the interactive manipulation of physical parameters

CAMP-G

The Computer Aided Modeling Program (CAMP-G) has been designed as a specialized bond graph modeling software, which has the tools to create models of physical systems starting with a bond graph model and ending up in source code form suitable for simulation languages [2]. This approach has led to the development of CAMP_G as a preprocessor for powerful simulation programs such as ACSL (Advanced Continuous Simulation Language) [7]. Recent developments have lead to the interface to MATLAB and SIMULINK [8], [9]. In each case CAMP-G has adapted to the format and standards of the Simulation Councils for the description on systems in symbolic form or to the programming requirements inherited in a programming language such as MATLAB. It has been developed to bring together those who understand bond graphs with those who understand other methods since it joins bond graph modeling and classical methods of simulation and control such as the approach discussed by Morris and Granda [10

SYSQUAKE

Computer-Aided Design (CAD) tools for automatic control are often based on a scripting language specialized for manipulating mathematical objects such as polynomials, matrices, and time- and frequency-domain series, and for creating plots. Their main advantage is the freedom they give to the user [11]. New designs can be tested easily and quickly, and collections of scripts (toolboxes) are available for many areas of automatic control. One of the most widely used software of this kind is MATLAB® [12].

In conjunction with these software programs, whose usefulness is obvious, there exists the need for more focused tools. The wish for more interactivity has resulted in tools often programmed in MATLAB.

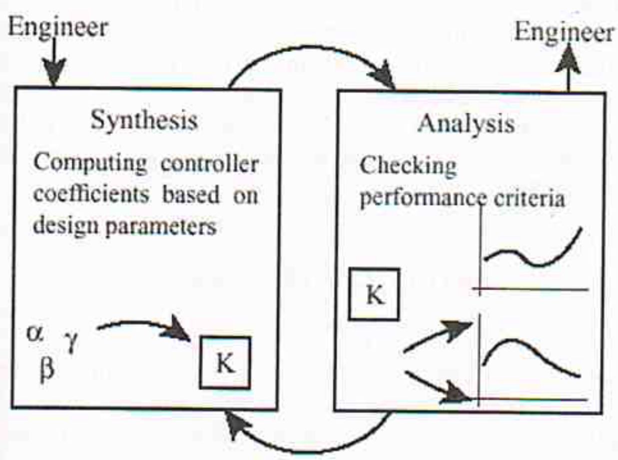

Figure 1: Standard design iterations between synthesis and analysis. K denotes the controller, and arbitrary design parameters.

For instance, one can mention the Quantitative Feedback Theory (QFT) toolbox [13], which permits the user easily to specify templates in the Nichols plot, and PARADISE [14], based on the Ackermann method [15] where the intersection of regions corresponding to some D-stability is sought graphically in the space of the controller parameters. However, software that enables manipulation of the design variables with instantaneous display of performance plots is still rare and here is where SysQuake comes in. From the perspective of the design tools, the history of the use of graphics as a help for controller design can be divided into three periods.

1.-Before the availability of digital computers. In addition to the calculation of a few numerical values, control engineers had sets of rules to draw graphics by hand.

2.-Computers make the creation of graphics much easier. Enabling fine-tuning of the design parameters by trial and error. The design process involves iterations between a phase of synthesis, where the coefficients of the controller **are** calculated from a set of design parameters, such that the closed-loop poles or the prediction window of a predictive controller; and a phase of analysis, where performance criteria are evaluated and compared to the specifications as shown in Figure 2.

3.-Interactive design with instantaneous performance display. This goes one step further. In many cases, it is not only possible to calculate the position of a graphic element (be it a curve, a pole or a template) from the model, controller and specifications, but also a new controller from the position of the element. For instance, a closed-loop pole can be computed by calculating the roots of the characteristic polynomial, based itself on the model and controller; and the controller coefficients can be synthesized from the set of closed-loop poles if some conditions on the degrees are satisfied. This two-way relationship between the graphic and the controller enables to manipulate the graphical elements with a mouse. Since a good design usually involves multiple objectives, several graphics are displayed and updated during the manipulation (see Figure 3).

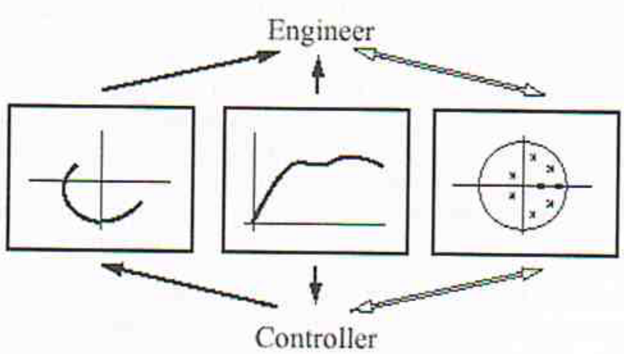

Figure 2: **SysQuake Interactive Design** with instantaneous performance display. The engineer manipulates one of the plots, which results in a new controller and updated graphics. The outlined arrow represents the manipulated graphic (in this case, the closed-loop poles).

Interactive design with instantaneous performance display offers two main advantages. First, it includes the engineer into a tight feedback loop of iterative design. The engineer can identify quickly the problems of his design and attempt to fix them. Second, and this is probably even more important, not only the effect of the manipulation of a design parameter is displayed, but its direction and amplitude become apparent. The engineer learns quickly which parameter to use and how to push the design in the direction of better satisfaction of the specifications. Fundamental limitations of the system and the type of controller are revealed [16], permitting a way to find an acceptable compromise between all the performance criteria.

Design involving a human being also benefits from a fast and intuitive approach because it lets him understand what happens. Boring search for the best set of parameters is avoided. And from a teaching point of view, effects deduced from simple systems and controllers can often be generalized and lead to an intuitive understanding of the underlying mathematics.

INTEGRATED ENVIRONMENT FOR OPEN LOOP SYSTEM UNDERSTANDING

The process illustrated by an example will take the reader from modeling to generation of C code ready to be implemented. The physical system is created using a bond graph model. It has been demonstrated in numerous research papers the ability of bond graphs to represent physical systems in multi-energy domains. In order to understand the principles involved, let's consider a simple suspension fourth order system shown in Figure 3 with its corresponding CAMP-G bond graph. The simple suspension system is discussed in the context of control in Reference 19 and shall here illustrate the procedure for open loop analysis for simulation purposes and close loop analysis for control purposes.

Once the bond graph on the right hand side of Figure 3 is entered and the SysQuake interface is selected, one obtains the SysqQuake model, which can have some default values, or the user can provide some for testing the system. The parameters can be changed simply by moving sliders so if the initial values are off, the response will show that.

Shown in Figure 4 is the step response, and the frequency response plots are shown simultaneously. On the right hand side, there is a set of sliders, which correspond to the physical parameters from the bond graph notation. Thus C2 for example is the Inertia element attached to bond 2. The engineer moves the sliders right or left and immediately sees the change in the step response and simultaneously and instantly the change in the frequency response.

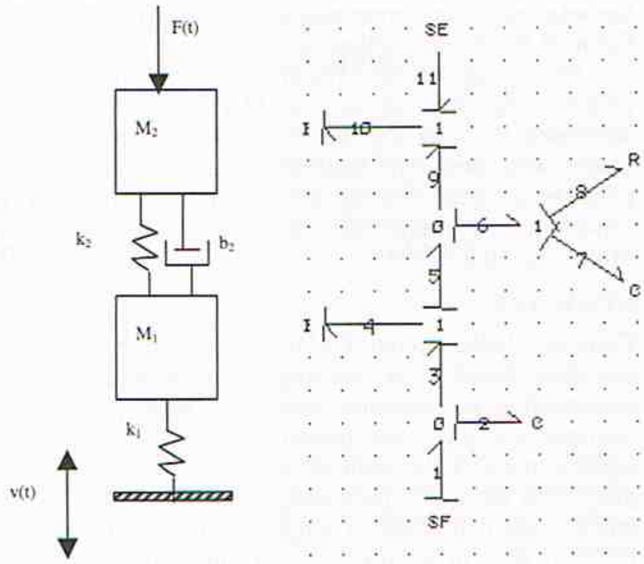

Figure 3. Physical System and Bond Graph

The sliders at the right hand side of Figure 4 are the parameter values from the bond graph model. The system of equations, which is underneath completely transparent to the user, is compatible with the format for SysQuake. The parameters can be changed even driving the system unstable. The closed loop menu allows studying the same system in close loop form so that a root locus, frequency response and a step response are computed simultaneously on the same screen to see the immediate change. The poles of the root locus can be moved and instantaneously see the gain and phase.

SIMULATION, CONTROLLER DESIGN

Granda and Fitsos [17] discuss the design of an electromechanical actuator and verified the mathematical equivalence of a classical approach of equation derivation and Laplace transform application with that of the bond graph method. Further more Granda [18] demonstrated how the method could be applied to the computer generation of transfer functions. These tools have laid the groundwork for the main argument on this paper, which is the generation of a model from a bond graph, which will be suitable for instantaneous synthesis and analysis. When designing a control system such as a PID control, the same procedure explained in the previous section is used to produce the state space form or the transfer functions as computer generated strings. Using the SYSQUAKE display shown on Fig 6 illustrates how the designer could control the changes to be made to fulfill the design specification.

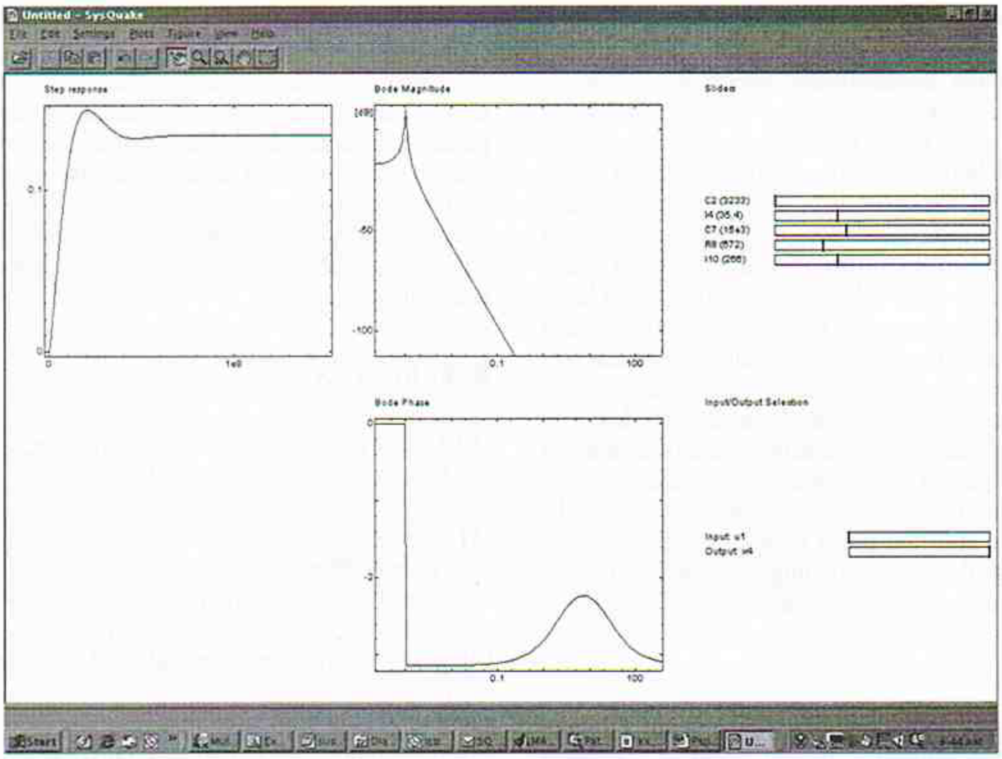

Figure 4. Open loop system analysis

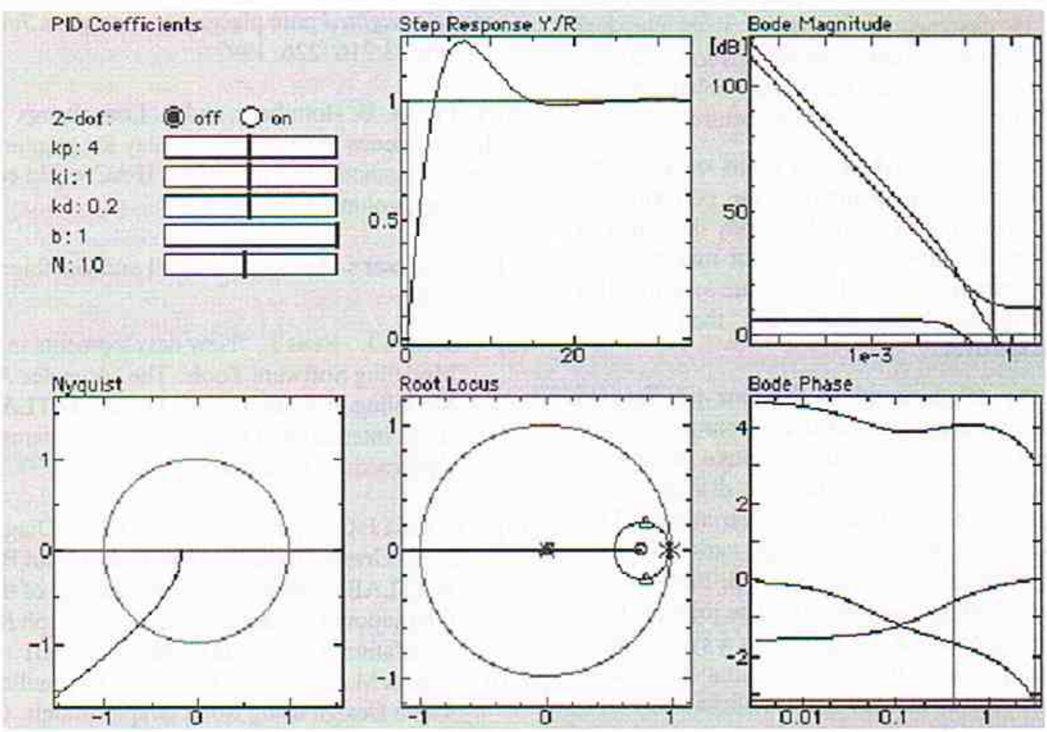

Figure 5 SysQuake Controller Design Panel

The previous section demonstrated the process of model generation and interface to SysQuake. It is now important to emphasize on SysQuake in order to profit from the method proposed here. The idea is that the bond graph model now would be analyzed with the tools within SysQuake, specialized tools for the understanding of systems and design of controllers. Once the bond graph model has been created, CAMP-G generated the SysQuake model in source code form and integrates the environment generating a model with some default values. The CAMP-G/SysQuake environment generates a workspace in SysQuake, which looks like the desktop display shown in Figure 5. This display puts the engineer in total control of the systems response in a graphical way so that physical parameters can be changed simply moving sliders, but the engineer can also grab the poles of a root locus plot for example to find the corresponding parameters or the instantaneous response of the system all done completely simultaneous.

CONCLUSIONS

This research has shown how an integrated software environment for modeling, for understanding of a systems as an open loop as well as closed loop controller design can be accomplished starting with a bond graph model. This approach enhances the power of bond graph software tools such as CAMP-G and allows the bond graph modeler to use software which is designed to be used with classical methods. This of course allows joining several methods with the same software but moreover enhances the understanding of a dynamics system and its control.

SysQuake also reveals the fundamental limits of a system, which may lead to modify it to improve the performance, which may be obtained. If it is desired to study the effect of changing the system by adding subsystems or making more severe changes to the structure of the dynamic system, all it takes is to modify the bond graph and then CAMP-G/SYSQUAKE takes over.

In this research the development of the new interface to SysQuake has produced an integrated environment with a very special objective in mind, the intuitive design of control systems by means of instantaneous display of the response of systems to physical parameter variations. This approach does not allow only narrow range variations such as in the study of sensitivity but to a full-blown scale changes to meet the design requirements. The joint package can also be used to understand the "plant" as a system in an open loop manner. However the study of it in a closed loop form with a controller is an extension a few clicks away.

This approach uses both synthesis and analysis methods in a simultaneous display because one can see the response to a parameter variation, but at the same time one can manipulate the response, to examine what the physical parameters should be. This is a completely new concept which will give those already empowered bond graphers who can produce computer generated models within minutes with the ability to study the bond graph model in state space form or in transfer function form using synthesis and analysis.

REFERENCES

[1.] I. Horowitz. Survey of quantitative feedback the*ory. Int. J. Control, 53(2):255–291, 1991*

[2] - Granda J. J. "Computer Generation of Physical System Differential Equations using Bond Graphs." Journal of the Franklin Institute, January/February Issue, 1985

[3] Y. Piguet. *SysQuake User* Manual. Calerga, Lausanne, 1999.

[4] M. Dimmler and Y. Piguet. Intuitive design of complex real-time control systems. In Rapid *Sys-tem Prototyping 200*0, page (submitted), Paris, 2000.

[5] Y. Piguet, U. Holmberg, and R. Longchamp. Multi-model *weighted* pole placement. European *Journal of Contro*l, 3:216–226, 1997

[6] Y. Piguet, U. Holmberg, and R. Longchamp. Instantaneous performance display for graphical control design methods. In Proc. of the IFAC world congress, Beijing, volume L, pages 403

[7] ACSL user's Manual. Mitchell and Gauthier Associates 1998.

[8] - Granda J. , Reus J. "New developments in Bond Graph Modeling Software Tools: The computer Aided Modeling Program CAMP-G and MATLAB". The IEEE International Conference on Systems, man, and Cybernetics. Orlando, Fla. October 1997.

[9] Granda J. Computer Generated Block Diagrams from Bond Graph Models CAMP-G as a Tool Box for MATLAB/SIMULINK.. Proceedings of the International Conference on Bond Graph Modeling and Simulation ICBGM'2001. Phoenix, 2001

[10] Morris M, Granda J. Four Way Hydraulic Control Valve Design using Bond Graph Models, Computer Generated Block Diagrams and SIMULINK S-Functions .. Proceedings of the International

Conference on Bond Graph Modeling and Simulation ICBGM'2001. Phoenix, 2001

[11] C. M. Rimvall and C. P. Jobling. Computer-aided control systems design. In W. S. Levine, editor, *The Control* Handbook, pages 429–442. CRC Press, 1996.

[12] The MathWorks, Inc., Natick, Mass. MATLAB, *the Language of Technical* Computing — *Using MATLAB*, 1997.

[13] O. Yaniv, Y. Chait, and C. Borghesani. The QFT control design toolbox for MATLAB. In *Proc. of* 2nd IFAC Symp. on Robust Control Design, pages 107–112, Budapest, 1997.

[14] W. Sienel, J. Ackermann, and T. B¨ unte. Design and analysis of robust control systems in PARADISE. In Proc. of 2nd IFAC Symp. on Robust *Control Design*, pages 71–76, Budapest, 1997.

[15] J. Ackermann, A. Barlett, D. Kaesbauer, W. Sienel, and R. Steinhauser. *Robust Control: Systems With Uncertain* Physical *Parameter*s. Communication and control engineering series. Springer, 1994.

[16] M. M. Seron, J. H. Braslavsky, and G. C. Goodwin. *Fundamental Limitations in Filtering and Control. Communication and control engineering series.* Springer, 1997.

[17] Fitsos P., Granda J. "Bond Graph Modeling of Engine Valve and Control System Using Electromechanical Rotary Actuators". Proceedings of the International Conference on Bond Graph Modeling and Simulation ICBGM'99. San Francisco, Ca. January 1999. Pg. 353

[18] Granda J. " Computer Generated Transfer Functions CAMP-G: Interface to MATLAB and SIMULINK" Proceedings of the International Conference on Bond Graph Modeling and Simulation ICBGM'99. San Francisco, Ca. January 1999. Pg. 129

[19] Karnopp, Margolis, Rosenberg "*System Dynamics: Modeling and Simulation of Mechatronics Systems*". Wiley and Sons, 2000.