Katholieke Universiteit Leuven

Departement Elektrotechniek ESAT-SISTA/TR 1997-56

BondLab, a Matlab Based GUI for Bond Graph Modeling

Willem Minten, Sven Vranckx, Bart De Moor, and Joos Vandewalle[1]

Published in a Special Issue of JournalA, Vol 38, No3, 1997

[1]ESAT (SISTA) - Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, 3001 Leuven (Heverlee), Belgium, Tel. 32/16/32 17 09, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuven.ac.be/sista*. E-mail: {Willem.Minten,Bart.Demoor,Joos.Vandewalle}@esat.kuleuven.ac.be.
Willem Minten is Research Assistant, Sven Vranckx is Graduate Student, Bart De Moor is a senior Research Associate with the F.W.O. (Fund for Scientific Research-Flanders), and Joos Vandewalle is Full Professor, all at ESAT.
This work is supported by several institutions: the Flemish Government (Concerted Research Action GOA-MIPS, the FWO (Fund for Scientific Research - Flanders: G.0292.95, G.0256.97), the FWO Research Communities (ICCoS, Advanced Numerical Methods for Mathematical Modeling), the IWT (Flemish Institute for Science and Technology in Industry: IWT/VCST project (1992-1997), IWT Action Program on Information Technology (ITA/GBO/T23) - IT-Generic Basic Research: ISIS)), the Belgian Government (Inter-university Attraction Poles: IUAP P4-02 (1997-2001) and IUAP P4-24 (1997-2001)), and the European Commission (Human Capital and Mobility Network SIMONET; SCIENCE-ERNSI SC1-CT92-0779; Keep In Touch SYSIDENT-KIT124).
The scientific responsibility is assumed by its authors.

BondLab: A Matlab Based GUI for Bond Graph Modeling

Willem Minten, Sven Vranckx, Bart De Moor, and Joos Vandewalle

Abstract: `BondLab`, an acronym for Bond Graph Laboratory, has been set up as a toolbox in Matlab (release 5) for mechatronic systems design. In this environment one can seamlessly connect both behavioral and causal model description formalisms that occur in the design of mechatronic systems. This software tool integrates the whole design process, from conceptual modeling to product design, in which each step towards the final product is integrated in the global design procedure, hence generating a potential breakthrough in the industry, where engineers are confronted with presently unsupported complex design objectives and closer deadlines.

Keywords: physical systems modeling, mechatronic design, interactive programs.

1. INTRODUCTION & PROBLEM STATEMENT

These days design engineers are confronted with a vast amount of software tools, each of which solves a specific aspect of the design cycle (Figure 1). Todays software aids are characterized by their *sequential* nature, i.e. the design is split up into separate sequential events, in which a later event (e.g. a controller parameter tuning) does not influence earlier events (e.g. the structure determination of the physical system).

1.1. Classical Design

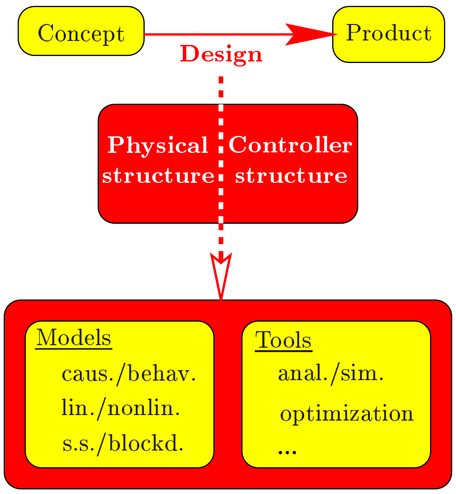

Figure 1: View on a product design cycle initiated by a given concept in mind. The design concepts in the conceptual phase *are totally different from those in the* product phase. *Consequently this results in different* models *and* tools *throughout a design cycle.*

In the conceptual design phase (Figure 1) proper models are behavioral. Only the physical aspects of how the system behaves (or should behave) are of interest. Such models tend to be nonlinear. With Bond Graphs as behavioral modeling tool, the modeler focuses on power and energy interactions between different parts of the model of the physical system, and not on the constraints of certain mathematical formulations of the problem statement. In general, the use of Bond Graphs results in better (i.e. right) models. In the product design phase, such behavioral models are to be transformed into tractable forms, i.e. causal models such as a (linear) state space form (for system analysis or controller design related objectives), or as a true nonlinear block diagram (for proper simulation in a fine tuning process of the free controller parameters). Hence several kinds of models describing the same system are used in a design cycle. Summarizing, it can be said that the character of the model changes according to the boundary conditions in which it is used.
Dedicated software tools exists [4, 2, 3, 7] to help the designer in this sequential design process, and most of the time the integration of the available tools is achieved by means of ASCII file transfer between the successive software programs. In this text such a design is called *classical*.

1.2. Mechatronic Design

In a *mechatronic* design framework the designer tries to focus on the integrated ensemble of the final design, rather than on specific sequential problems in it. At first glance this complicates matters because design constraints need to be considered simultaneously. However, with the aid of proper modeling languages and software support, this integrated approach results in superior designs. Indeed, it can be shown that the more relevant constraints (on both the physical system and the controller) and relevant design degrees of freedom (e.g. originating from a controller to be implemented) are used, especially in the initial stages, the better the design meets the specifications:

1. The designer encounters fewer backtracking traces in her search for the optimal design.
2. The design time is considerably reduced because no design efforts are spoiled to comply with specs, which, at the end, seem hard to achieve.
3. The final design is superior to classical ones since the design effort is focussed towards all the relevant degrees of freedom in the design space, which on his hand is already constrained properly.

This implies that controller related specs are not only taken into account after the physical system has been designed and optimized. Again here Bond Graphs can aid. Using this modeling language the designer gets insights in the interacting parts of the conceptual model, previously to any model simulation [8]. Hence, the analysis of the models is not reserved for the last stages of a design alone, as is the case with classical design methods. Obviously, for a mechatronic design there is a need for a new tool that integrates conceptual design and product design objectives. Such an integration is only possible if the different models *and* the tools are also integrated (Figure 1). *BondLab* is an implementation of such a mechatronic design environment in Matlab5.
BondLab solves mechatronic problems in a very elegant way:

- The model entry is supported by a highly perceptive Graphical User Interface (GUI) (Section 2). This GUI also supports the investigation of the model quality with respect to the final desired design (Section 4).
- A first corner stone in an efficient design environment is the ability to automate the subsequent model transformations throughout a design (Section 3).
- A second corner stone involves the investigation of the model quality with respect to the final product (Section 4). Also the learning support of the designer, a largely underestimated item in actual software tools, is to be addressed here.

All this will be explained by means of an example, the physical model of which is shown in Figure 2 [9].

2. THE USER INTERFACE

Figure 3 shows the main window of BondLab in which a transformer is selected. In order to minimize the manipulations during the creation or modification of the Bond Graph by the user, several push-buttons are provided under the pull-down menus. Keyboard shortcuts for creation and modification purposes are also available (*move, copy, cut, paste, delete, flip horizontal and vertical, and rotate*). In order to prevent modeling errors BondLab is designed according to the standards described in [11]. The interface is extremely user-friendly [13, 14], and the BondLab database is designed for speed and code transparency [12]. The result is a compact, powerful and highly perceptive interface in which the designer feels comfortable (For example, the number of windows is restricted to the minimum). Below the graph there is a possibility to enter or change the most important properties of the selected element. Also nonlinear behavior can be modeled in several ways, since Matlab code can be entered right away in the value field of the elements. Even user defined .m function files can be used in the same straightforward manner. Using a properties window (shown in the figure) it is possible to have direct access to the whole database.
While (or after) drawing the Bond Graph, parameter labels

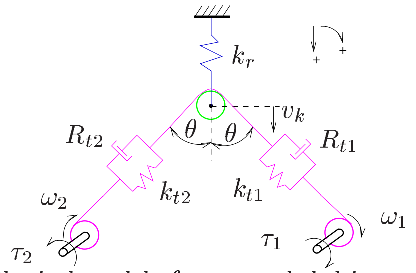

Figure 2: Physical model of two coupled drives, the drive side is located on the right hand side. A tape (index t) is transported from the left to the right. The tape tension is partially determined by the spring k_r.

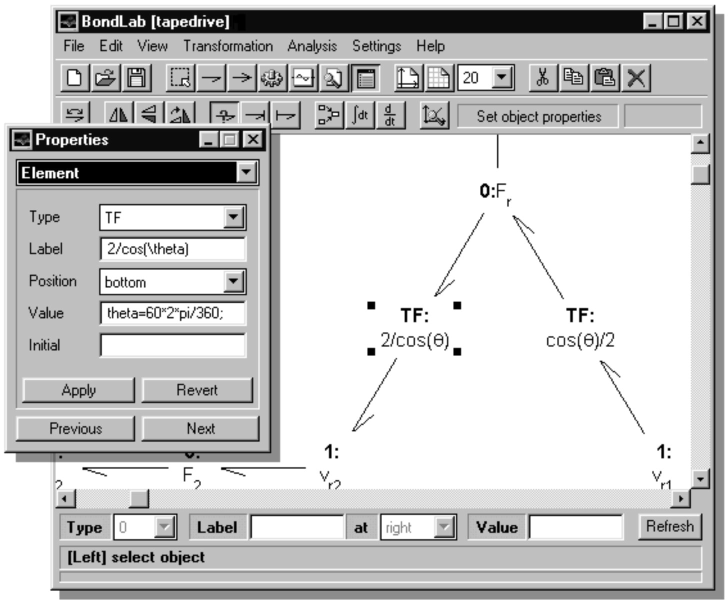

Figure 3: Screen-shot of the main window of BondLab in Properties Mode (reduced color-map). The properties window of the selected element is also shown.

and values can be entered, and a simplification routine can be invoked to remove e.g. redundant junction structure constructs. Thereafter a consistency check can modify questionable Bond Graph parts [10] and remove disconnected elements. Finally, upon designer demand, the causal assignment routines assigns the element statements from the equations.

3. AUTOMATIC MODEL TRANSFORMATIONS

Several causal analysis procedures can be invoked. The two most important ones are named SCAP (for derivation of the set of first order differential equations, and other related explicit forms like transfer functions, block diagrams, ...) or D-SCAP (for system inversion and additional Bond Graph related analysis). A symbolic or numerical output can be saved to the Matlab workspace from the *Transformations Tab* in the *Analysis* window, shown in Figure 4.

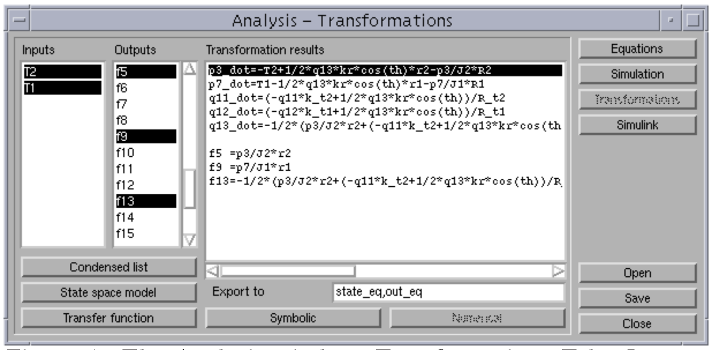

Figure 4: The Analysis window, Transformations Tab. Input and output variables can be chosen freely from the two lists on the left. The state and output equations and the transfer functions are calculated accordingly. The push-buttons in the lower middle of the window are used for symbolic or numerical export of the equations into the Matlab workspace.

This output can be a (non-)linear state space form or transfer matrix form, shown in Table 1.

Remark that all this is integrated in Matlab in a flexible way, e.g. LaTeX code [5] can accompany the element labels resulting in very readable graphs (The LaTeX mark up code is removed automatically when a symbolic form of the equations is desired), and the derived equations can be visualized as LaTeX code or as Matlab relevant code.

BondLab also provides internal facilities for direct numerical simulation. Therefore, the set of ODE's can be simulated from within the *Simulation* Tab of the *Analysis* window, shown in Figure 5.

If unavoidable derivative causalities occur, or if some unsolvable nonlinear algebraic loops appear, then BondLab can automatically create an optimal Simulink `.mdl` block diagram, which consequently can also be simulated.

4. MECHATRONIC DESIGN SUPPORT

At last two basic tools will be explained shortly.

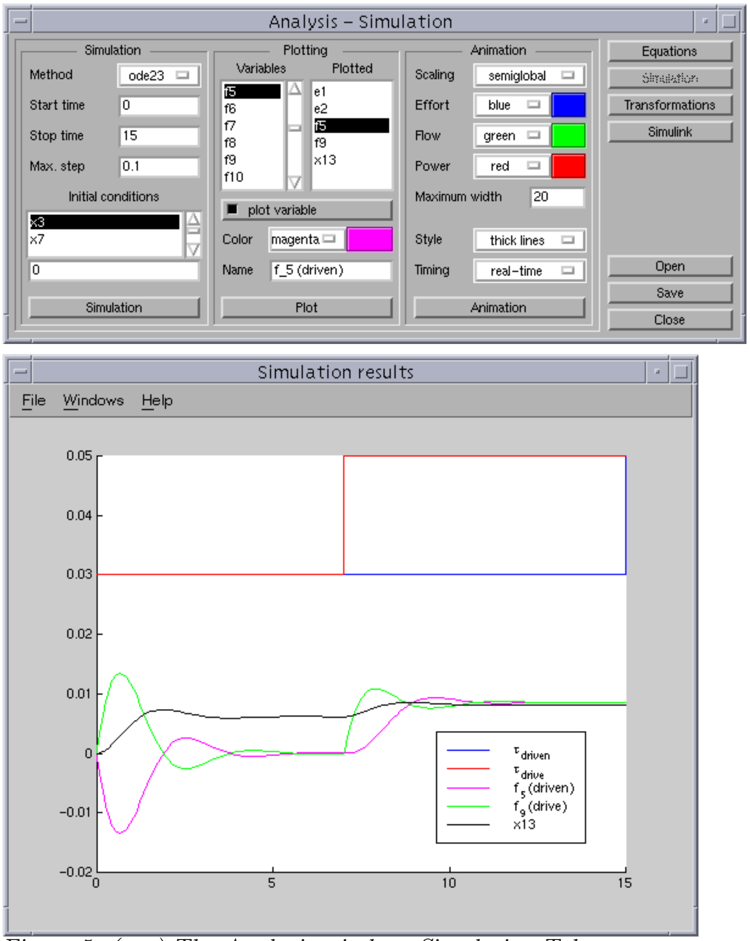

Figure 5: (top) The Analysis window, Simulation Tab. (bottom) Simulation results of the coupled drives: The tensioning of the tape ($0 - 7sec$) and the roll up ($7 - 15sec$) simulation results are visualized in this plot window.

4.1. Energy based model reduction

After simulation, the power distribution in the Bond Graph can be visualized in the Bond Graph [9, 6]. BondLab achieves this e.g. by changing the width of the bonds according to the instantaneously transmitted power. This is especially useful for the reduction of nonlinear systems, in which the nonlinear behavior in time as function of an input shape should be evaluated. In Figure 6 a snap-shot of such an animation based analysis is shown. In this example it turned out that 96% of the power-picture remains the same when the tape dynamics are neglected. This observation makes it possible to reduce the Bond Graph from a fifth to a third order system, while maintaining the most important dynamics. For this (linear) model it turns out that the fastest poles are omitted in this reduction step. Remark that the power of this model reduction approach originates from the fact that it has been executed in the physical domain and hence, does not depend on a certain mathematical description of the problem. Therefore this method can directly be applied to nonlinear systems in which e.g. the

$$
\begin{bmatrix} \dot{p}_3 \\ \dot{p}_7 \\ \dot{q}_{11} \\ \dot{q}_{12} \\ \dot{q}_{13} \end{bmatrix} =
\begin{bmatrix}
-\frac{R2}{J2} & 0 & 0 & 0 & \frac{kr\ \cos(th) r2}{2} \\
0 & -\frac{R1}{J1} & 0 & 0 & -\frac{kr\ \cos(th) r1}{2} \\
0 & 0 & -\frac{k\_t2}{R\_t2} & 0 & \frac{kr\ \cos(th)}{2\ R\_t2} \\
0 & 0 & 0 & -\frac{k\_t1}{R\_t1} & \frac{kr\ \cos(th)}{2\ R\_t1} \\
-\frac{r2\ \cos(th)}{2\ J2} & \frac{r1\ \cos(th)}{2\ J1} & \frac{k\_t2\ \cos(th)}{2\ R\_t2} & \frac{k\_t1\ \cos(th)}{2\ R\_t1} & -\frac{kr\ (\cos(th))^2}{4}\left(\frac{1}{R\_t2}+\frac{1}{R\_t1}\right)
\end{bmatrix} \cdot
\begin{bmatrix} p_3 \\ p_7 \\ q_{11} \\ q_{12} \\ q_{13} \end{bmatrix} +
\begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot
\begin{bmatrix} T_2 \\ T_1 \end{bmatrix}
$$

$$
\begin{bmatrix} e_9 \\ f_9 \end{bmatrix} =
\begin{bmatrix} 0 & 0 & 0 & 0 & 1/2\, kr \cos(th) \\ 0 & \frac{r1}{J1} & 0 & 0 & 0 \end{bmatrix} \cdot
\begin{bmatrix} p_3 \\ p_7 \\ q_{11} \\ q_{12} \\ q_{13} \end{bmatrix} +
\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot
\begin{bmatrix} T_2 \\ T_1 \end{bmatrix}
$$

$$
\frac{f_9}{T_1} = \frac{num(s^4)}{den(s^5)} = \frac{51/100 * (s+10) * (544 * s^3 + 6522 * s^2 + 10742 * s + 9945)}{17/400 * (s+10) * (4 * s + 7) * (1088 * s^3 + 13044 * s^2 + 23473 * s + 39780)}
$$

Table 1: State space form (symbolic) and transfer function description (numerical) of the Bond Graph of Figure 2. These equations (LaTeX format) are copied straight from the Matlab workspace.

input shape dependency has to be taken into account.

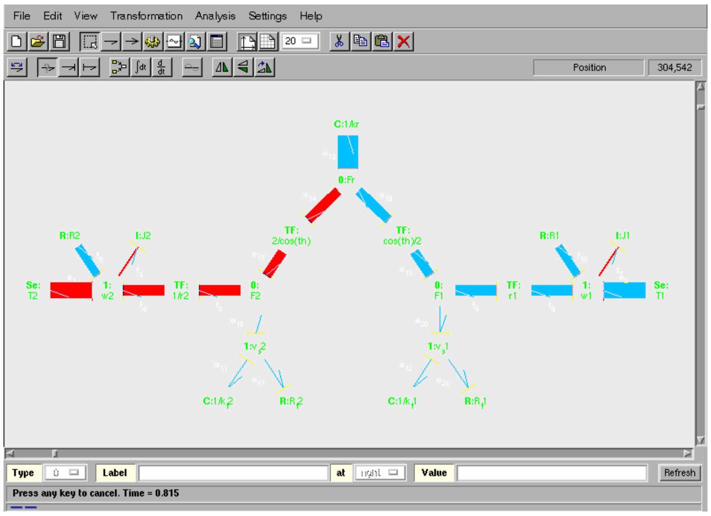

Figure 6: Snap-shot at 0.816sec of the animation of the energy flow. This figure shows the power transmitted in each bond. The width of each bond is proportional to the instantaneously transmitted power. A red (dark) line depicts negative power with respect to the power arrow reference, a blue (bright) one depicts positive power.

Classical methods (i.e. by means of minimal rank approximations with singular values or by means of eigenvalue decomposition) are not appropriate for this.

4.2. Structural properties of a model to be controlled

BondLab also provides structural aids for investigating model quality with respect to final controlled system specs. The explicit use of causal analysis reveals relevant structural system properties [8], previous to any model transformation or simulation. For linear systems a lot of effort has already been done [1]. It is a pity that these results are not implemented in a software tool, accessible for industry. As an illustration we outline shortly the control of the variator of a computer controlled CVT (Continuously Variable Transmission). In Figure 7 the reduced physical model and the general Bond Graph of the variator are depicted. One of the design questions is to look for the optimal actuation of this variator. Three controller configurations are of interest: a `pp` (i.e. S_p and S_s are pressure sources), a `pQ` (i.e. S_s is a pressure source and S_p is a `Q` (flow) source), or a `QQ` configuration. From simple causal path analysis of the Bond Graph it turns out that the `pQ` configuration (see Figure 7 (bottom)) is superior with regard to the decoupling aspects of the multivariate controlled system, despite its higher order in comparison with a `pp` configuration (5 instead of 4). Also, it can be seen that the quality of the push-belt spring k_{pb} (and to a lesser extend the mass of the secondary pulley m_s) determines the robustness of such a `pQ` configuration. If this configuration is used, the behavior of the secondary pressure source Se should be quite ideal, hence the physical system designer has to pay attention to this source. All these qualitative results steer the design in the good direction. Of course additional simulations are necessary in order to retrieve quantitative data.
Notice that the causal investigation of such a Bond Graph helps the designer to improve his design skill for further designs.

5. CONCLUSIONS

`BondLab` is an integrated tool for the design of mechatronic

products. Its flexibility and user-friendliness are two of its most appreciated aspects. The used concepts are based on Bond Graphs, which makes it possible to exploit the nature of physical systems (behavioral) and the human perception (graphs). Other functionalities include an automatic computer support (model transformations), and the ability to interfere or interrogate the models with regard to the final design (mechatronics). The seamless implementation of this tool is done as MatLab5 `.m` files, which makes this tool platform-independent (PC, UNIX workstation, ...).
If required to do so, we can provide teaching and training of Bond Graph methods in which BondLab play a central role.

References

[1] Coutereel L., Bouayad R., Dauphin-Tanguy G., (1989), "Artificial Intelligence and Bond Graph Methodology for the Modeling of Dynamical Systems", IMACS Transactions Scientific Computing '88, Vol.3: Modeling and Simulation of Systems, J.C. Baltzer, Swiss.

[2] Elmqvist H., "DYMOLA Homepage", Dynasim AB, Lund, Sweden, `http://www.Dynasim.se`, (1996).

[3] Elmqvist H., "MODELICA Homepage", `http://www.Dynasim.se/Modelica`, (1996).

[4] Kleijn C., "Twente Simulation Homepage", Controllab Products B.V., `http://www.rt.el.utwente.nl/20sim/clp.htm`, (1996).

[5] Lamport L., "LaTeX: a document prepation system", 2nd Edition, Addison-Wesley Publishing Company, (1995).

[6] Lorenz F., Goffart M., "Direct representation of Simulation results on the Bond Graph Model", *Proc. of the 3rd International Conference on Bond Graph Modeling and Simulation, Part of the WMC'97 Multiconference (ICBGM'97)*, Phoenix, Arizona, Jan., pp. 20-25. (1997).

[7] The Mathworks Inc, "Matlab release 5", 24 Prime Park Way, Natic, MA 01760-1500, USA, `http://www.mathworks.com`, (1996).

[8] Minten W., Vanvuchelen P., Moons C., De Moor B., "Modeling of the CVT in a virtual engineering environment", in Proc. of the 2nd International Conference on Road Vehicle Automation (ROVA'95), Bolton, UK, Sep., pp. 161-170, (1995).

[9] Minten W., De Moor B., and Vandewalle J. "Power and Power Variables in Graphical Modeling of Physical Dynamic Systems", ESAT-SISTA, report TR96-92, K.U.Leuven, Belgium, (1996).

[10] Minten W., De Moor B., and Vandewalle J. 1996 "How Can Different Interpretations and Use of Bond Graph Parameters be Avoided.", *J. Franklin Institute*, Vol334b, No3, pp.389-406, (1997).

[11] Minten W., De Moor B., Vandewalle J., "Condensed Bond Graph modeling", *Proc. of the 3rd International Conference on Bond Graph Modeling and Simulation, Part of the WMC'97 Multiconference (ICBGM'97)*, Phoenix, Arizona, Jan., pp. 50-55. (1997).

[12] Minten W., Vranckx S., De Moor B., Vandewalle J., "Bondlab, a Toolbox for Integrated Mechatronic Design", 11th European Simulation Multiconference (ESM97), Istanbul, Turkey, July, pp 199-203, (1997).

[13] Preece J., Keller L., *Eds*, *Human-Computer Interaction*, Prentice Hall, UK, (1994).

[14] Tognazzini B., *Tog on Interface*, Addison Wesley, New York, USA, (1991).

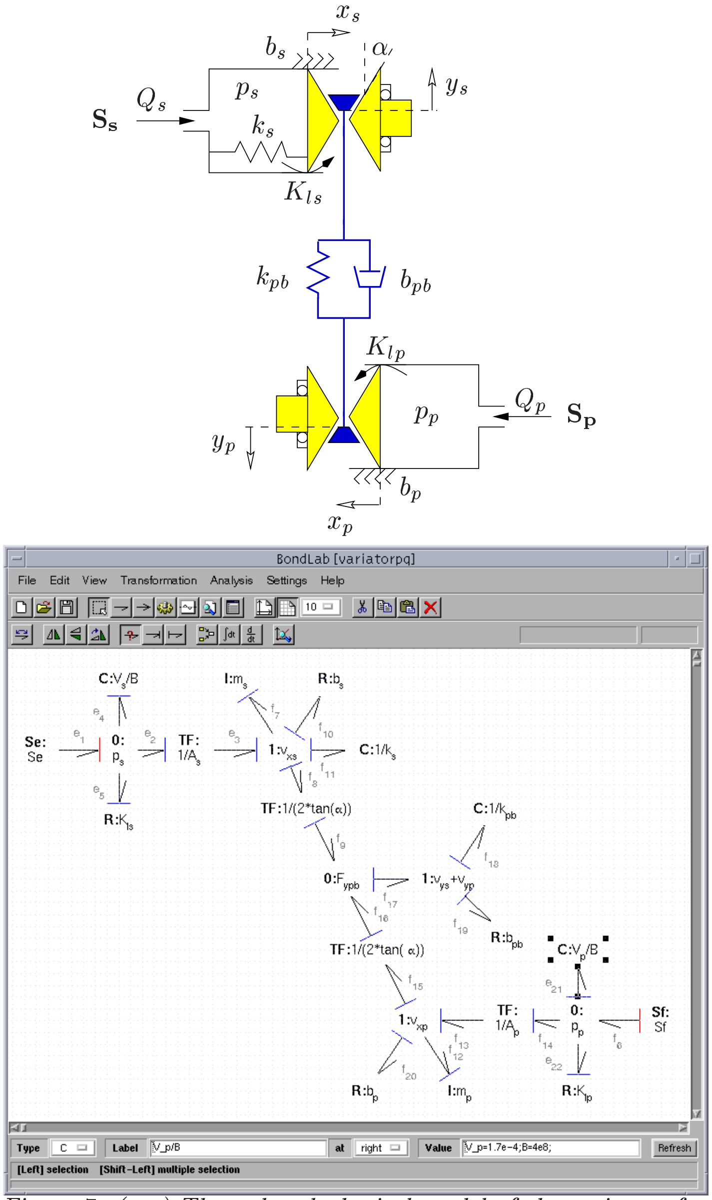

Figure 7: (top) The reduced physical model of the variator of the CVT. The primary side (index p) is located at the engine side of the transmission, the secondary side (index s) is located at the wheel side. A reduced model of the push-belt is also shown (index pb).
(bottom) The main window of BondLab depicting the general Bond Graph of the variator in a `pQ` *configuration.*