Program for Multi-Domain Symbolic Analysis

Zdenek Kolka, Dalibor Biolek, Jaroslav Kalous, Viera Biolkova
Faculty of Electrical Engineering and Communication
Brno University of Technology
Brno, Czech Republic
kolka@feec.vutbr.cz

Abstract— **The paper deals with the implementation of multi-domain symbolic analysis in a new program, SAMD (Symbolic Analysis of Mechatronic Drives), developed with the emphasis on linearized hybrid systems combining classical electrical circuits, controllers, electro-mechanical converters and mechanical parts of electric drives. The systems analyzed can be composed of both basic elements and more complex blocks. All models are stored in an easily extensible library. Implemented algorithms for equation/matrix-based symbolic simplification allow analyzing larger systems.**

Keywords-symbolic analysis; mechatronics; multiple-domain linear systems; symbolic simplification

I. Introduction

Mechatronic systems integrate electrical, mechanical and control blocks into a complex system [1]. It is preferable to use one simulation platform for the whole system instead of using several specialized programs, each one for one physical domain. The problem of modeling lumped mechatronic systems can be solved via three different approaches:

1) Utilization of a tool specialized in one physical domain for the modeling of multi-domain systems on the basis of analogies. A typical example is the use of SPICE simulators with equivalent circuit models.

2) Utilization of universal program tools for scientific computation such as MATLAB, MATHEMATICA, MathCAD, Maple, etc. A typical case is the utilization of MATLAB/Simulink with SimElectronics, SimPowerSystems, SimMechanics, SimDriveline, and SimScape toolboxes.

3) Utilization of tools developed for the modeling and simulations of large systems of general nature. The class contains various VHDL-AMS and Verilog-AMS simulators, and other tools like ViSSim, Portunus, Saber, Dynast, etc.

All the above-mentioned simulators provide numerical solutions. Their internal algorithms are based on classical methods for the solution of a set of algebraic-differential equations. Several programs provide semisymbolic solutions [2], i.e. polynomial coefficients or poles and zeroes, which are applicable to linear or linearized models.

Lumped linear or linearized systems can be solved completely symbolically. There are many programs from group (1) for the analysis of electrical circuits. The commercial SPICE simulator TINA can generate symbolic results. Additionally, a number of free software programs are available. Let us mention SAPWIN [3], SIASCA, and others. SNAP is a popular program used for the symbolic and semisymbolic analysis of linearized circuits [4], [5]. In other technical fields, the symbolic analysis is rarely used.

Universal programs from group (2) can perform symbolic computations as well. MAPLE is based on symbolic algorithms and the analysis is also available in MATLAB with the Symbolic toolbox. A special package, Analog Insydes, running within MATHEMATICA is commercially available.

The symbolic analysis of models of real systems can lead to very complex formulae, whose practical utilization is questionable or even doubtful. In recent years, considerable effort was devoted to the development of the approximate symbolic analysis of electrical circuits, which is able to simplify symbolic formulae at the cost of loss of their general validity [6].

The paper deals with the implementation of multi-domain symbolic analysis in a new program, SAMD (Symbolic Analysis of Mechatronic Drives), developed with the emphasis on hybrid systems combining classical electrical circuits, controllers, electro-mechanical converters and mechanical parts of electric drives [7]. The basic symbolic algorithms are based on those used in the SNAP simulator. Our original algorithm [8] of Simplification Before Generation (SBG) is implemented in SAMD. The program has been developed with the aim of avoiding the use of tedious and error-prone analogies. It can be used both by professionals and for education. Section 2 of the paper deals with the methods and algorithms used, and Section 3 provides a practical example of the symbolic analysis of a complex system and also shows the potentials of symbolic simplification.

II. Symbolic Analysis of Mechatronic Systems

A. Basic concept

The modeling strategy is based on a general principle that a lumped system to be analyzed can be decomposed into individual parts. Each part represents either a real functional block (amplifier, regulator, gear box, motor, etc.) or an element (basic circuit element, mechanical element – shaft, disc, dumper, etc.). It is assumed that energy transmission between

This work has been supported by the Czech Science Foundation under grant No. P102/10/1665. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 230126.

978-1-4244-6815-7/10/$26.00 ©2010 IEEE

the parts runs through a small number of discrete terminals with associated *effort* (*across*) and *flow* (*through*) *power quantities* [11], Table I.

TABLE I. ANALOGOUS POWER QUANTITIES

Physical nature	Power quantities	
	Flow	***Effort***
electrical	current i [A]	voltage v [V]
translational	force F [N]	translational speed v [m s^{-1}]
rotational	torque T [N m]	angular speed ω [rad s^{-1}]

Traditionally, there are three[1] basic ideal elements represented by the three constitutive equations

$$f(t) = \alpha\, e(t)\,, \quad \int f(t)dt = \beta\, e(t)\,, \quad \int e(t)dt = \gamma\, f(t)\,, \quad (1)$$

where $f(t)$ and $e(t)$ stand for flow and effort quantities, respectively. The coefficients α, β, and γ are parameters of the elements, which are specified in Table II for individual kinds of physical nature.

TABLE II. PARAMETERS OF BASIC ELEMENTS

Physical nature	Parameter		
	α	β	γ
electrical	conductance G [S]	capacitance C [F]	inductance L [H]
translational	damping b [N s m^{-1}]	mass m [kg]	compliance d [m N^{-1}]
rotational	torsional damping b_t [N m s rad^{-1}]	inertia J [kg m^2]	torsional compliance d_t [rad N^{-1} m^{-1}]

The basic physical conservation laws for all three kinds of nature lead to the generalized Kirchhoff's laws for power quantities. As both the constitutive equations (1) and coupling conditions are formally the same, the set of linear equations describing the linear mechatronic system being modeled can be obtained using the generalization of the Modified Nodal Approach Analysis (MNA) [10].

Let us call the method the Modified Method of Effort Quantities (MMEQ), which leads to the set of equations in the general form

$$\mathbf{H}\mathbf{x} = \mathbf{b}\,, \qquad (2)$$

where **H** is the hybrid system matrix, **b** is the vector of sources, and **x** is the vector of unknown system variables.

In a simple case the **x** vector of unknown variables contains the effort quantities, i.e. electrical voltages for electrical systems and translational or angular speeds for mechanical systems. The effort quantities are of potential nature, i.e. they are defined relatively as differences of generalized potentials (electrical voltage is a difference of electrical potentials, the speed of the object is derived relative to the observer). In order to define the efforts unambiguously, we have to introduce the so-called reference quantity (reference), and relate all the efforts of the mechatronic system to it. For a computer analysis of hybrid systems, it is useful to define electrical and mechanical references separately.

The vector of sources **b** contains the flow quantities of the system, i.e. electrical currents for electrical circuits, and forces or driving moments and torques for mechanical systems. Starting from the analogy to electrical systems, the **H** matrix can be called the generalized admittance matrix of mechatronic system. This matrix can be obtained via composing successively the partial matrices of individual sub-systems, the so-called matrix-stamps [12].

In certain cases, when the flow cannot be derived as a function of the efforts, the generalized admittance matrix of the system is not defined. Then the vector **x** of unknown variables in (1) should be extended by properly selected flows, and the set of equations must be completed with the same number of equations which describe the relations between the augmented flows and the efforts of the system. This extended set of equations (1) then describes the Modified Method of Effort Quantities, and the corresponding hybrid matrix can be obtained via superposition of hybrid matrices-stamps of partial sub-systems.

B. Multi-domain models

The modeled system can be decomposed into both basic elements and more complex entities. This approach has been chosen to avoid unnecessary use of analogies. For example, an ideal block whose transfer function has a single real pole is usually modeled using R and C elements and two controlled sources. The product *RC* appears in the resulting symbolic formula. The modeling style implemented in SAMD allows defining a 1st-order transfer function with a single parameter, say τ, which simplifies interpreting the symbolic formula.

Let us give an example of the modeling of a permanent-magnet DC motor, Fig. 1. The motor transforms the electrical current I_a into the torque T_o, and, simultaneously, the rotation of the rotor induces electrical voltage in the armature winding.

The given electric drive is described by the following set of equations:

$$R_a I_a + L_a \frac{dI_a}{dt} + V_i = V_a\,, \qquad (3)$$

$$V_i = c_\Phi\, \omega_o\,,\; T_i = c_\Phi\, I_a\,, \qquad (4)$$

$$J_r \frac{d\omega_o}{dt} + T_o - T_i = 0\,, \qquad (5)$$

[1] Indeed, an infinite number of basic elements, including the famous Chua's memristor, can be defined analogously to (1) [9].

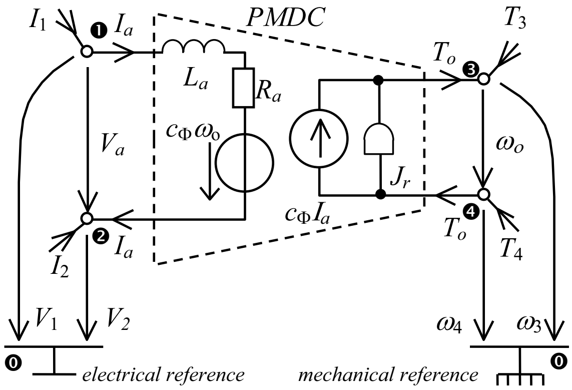


Figure 1. Model of permanent-magnet DC motor.

```
[PMDC]
terminals  = E E R R
params = 4
names  = cf,Ra,La,Jr
add    = 1
mat1   = 1 -1  2  0 : 1
mat2   = 3 -1  4  0 :-cf
mat3   =-1  1  0  2 : 1
mat4   =-1  4  0  3 : cf
mat5   = 3  3  4  4 : s*Jr
mat6   =-1  5  0  0 : [Ra+s*La]
```

Netlist call:
PMDC_M1 n1 n2 n3 n4 cf, Ra, La, Jr

Figure 2. Model in SAMD library and its netlist call.

where R_a and L_a are the resistance and the inductance of the armature winding, c_Φ is the torque constant (depends on motor design), J_r is the inertia of the motor rotor. The symbols I_a and V_a represent the armature current and voltage, V_i is the induced voltage, T_i is the electromagnetic torque, T_o and ω_o are output torque and angular speed.

In the most general case, the motor (model) is considered to be "floating", i.e. with the efforts defined as differences of electrical voltages or angular speeds, which relate to electrical or mechanical references, see Fig. 1.

The motor interacts with its neighborhood only at the interface points, marked in Fig. 1 by the symbols ❶❷❸❹. The corresponding interface quantities are voltages V_1 and V_2 and currents I_1 and I_2 for the electrical section, and angular speeds ω_3 and ω_4 and torques T_3 and T_4 for the mechanical section.

On the basis of (2)-(5) and Fig. 1, a set of MMEQ equations for the motor in the frequency domain can be found:

$$\left[\begin{array}{cc:c:c|c} & & & & 1 \\ \hdashline & & & & -1 \\ \hdashline & & sJ_r & -sJ_r & -c_\Phi \\ \hdashline & & -sJ_r & sJ_r & c_\Phi \\ \hline 1 & -1 & -c_\Phi & c_\Phi & R_a+sL_a \end{array}\right]\left[\begin{array}{c} V_1 \\ \hdashline V_2 \\ \hdashline \omega_1 \\ \hdashline \omega_1 \\ \hline I_a \end{array}\right]=\left[\begin{array}{c} I_1 \\ \hdashline I_2 \\ \hdashline T_3 \\ \hdashline T_4 \\ \hline 0 \end{array}\right] \qquad (6)$$

The winding current I_a was added to the set of unknowns. This addition must be accomplished by extending the set of equations by one additional equation (see the row below the solid line), which represents equation (3) for evaluating the voltage V_a.

Fig. 2 shows the library model for the motor and an example of a Spice-like netlist line. The stamp element P_i (from $mat_i = a\ b\ c\ d : P_i$) appears with the positive sign in positions (a,b) and (c,d), and with the negative sign in (a,d) and (c,b). Within the model, the interface terminals are numbered sequentially from 1. Terminal "1" corresponds to the first position in the netlist, i.e. local node "1" becomes a global node "n1" in the example in Fig. 2. The first augmented row and column are denoted "-1" while "0" means "no entry" to the matrix. The formula in square brackets will not be expanded during symbolic analysis.

The keyword `terminals` defines the kind of physical nature for each terminal (E-electrical, R-rotational, T-translational). If incompatible terminals are connected, the program reports an error. The keywords `params` and `names` define the number of model parameters and their local names. The `add` keyword defines the number of augmented quantities.

C. Symbolic algorithms

In the case of a mechatronic system driven by a single source it is possible to define four types of transfer function as response/excitation ratios, which correspond to four possible combinations of *efforts* and *flows*. The corresponding transfer function $F(s)$ can be expressed as a ratio of two algebraic cofactors of matrix $\mathbf{H}$

$$F(s) = \frac{response}{excitation} = (-1)^\alpha \frac{\det(\mathbf{H}_1)}{\det(\mathbf{H}_2)}, \qquad (7)$$

where $\mathbf{H}_1$ and $\mathbf{H}_2$ are submatrices obtained by means of adding and deleting some rows and columns of $\mathbf{H}$. The coefficient α depends on the indices of deleted rows and columns [13].

The expansion of (7) is theoretically straightforward, but it is practically meaningful only for systems where the final formula will be small and thus interpretable. Otherwise it is necessary to use the approximate symbolic analysis [6]. SAMD can also detect parallel or series connections of similar components, which can simplify the resulting formula significantly for certain topologies, see Section 3.

Two algorithms are implemented in SAMD: SBG and SAG. The original SBG algorithm is described in our paper [8]. It is based on a combination of parametric simplification (i.e. on removing individual parameters by setting their values either to zero or to infinity) in combination with graph transformation. Optionally, the user can also employ a standard sensitivity-based SAG algorithm [14].

III. Examples

The first example shows the capability of SAMD to perform lossless topological simplification. Fig. 3 shows a simple ladder network and SAMD output for the input impedance. The same formula in the expanded plain format contains 34 terms in the numerator and 21 terms in the denominator.

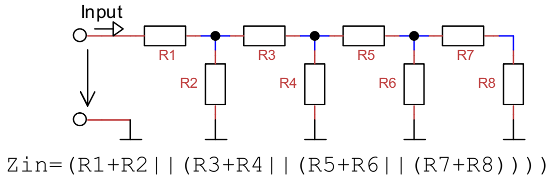
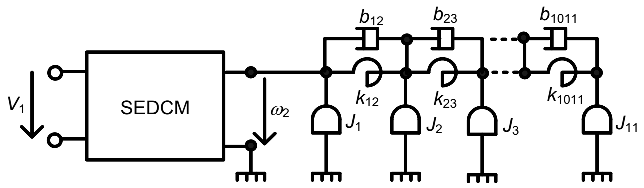


Figure 3. Ladder network.



motor: R_a=28.5m, L_a=0.66m, c_Φ=18.94
mechanical part: J_1=1140, J_2=62.5, J_3=21.9, J_4=16.4, J_5=15.9, J_6=40.7, J_7=51.3, J_8=18.6, J_9=16.2, J_{10}=200, J_{11}=22.4
k_{12}=34.2 10^6, k_{23}=19.3 10^6, k_{34}=11.8 10^6, k_{45}=20.9 10^6, k_{56}=59 10^6, k_{67}=20.9 10^6, k_{78}=58.3 10^6, k_{89}=15.5 10^6, k_{910}=32.5 10^6, k_{1011}=3.28 10^6, b_{12}=34.2 10^3, b_{23}=19.3 10^3, b_{34}=11.8 10^3, b_{45}=20.9 10^3, b_{56}=59 10^3, b_{67}=20.9 10^3, b_{78}=58.3 10^3, b_{89}=15.5 10^3, b_{910}=32.5 10^3, b_{1011}=3.28 10^3,

Figure 4. Model of vertical rolling mill stand.
(Physical units not shown due to limited space.)

The second example shows a model of rolling mill stand drive with a 1000 kW separately excited DC motor (equivalent to PMDC motor). The mechanical part is represented by 11 inertia elements (J_1 to J_{11}) connected with damped (b_{ij}) elastic shafts (k_{ij}). The shafts are represented by stiffness $k = 1/d_t$, see Table II. The model of the motor is shown in Fig. 1.

The transfer function

$$F(s) = \frac{\omega_2}{V_1} \tag{8}$$

is of the 22nd-order (i.e. it has 22 poles) and cannot be expressed symbolically due to its size. Using two reference points f_1 = 10 mHz and f_2 = 10 Hz with an allowed tolerance of 1 dB and 5 degrees the simplification algorithms found a 2nd-order transfer function

$$F_{app}(s) = \frac{1}{c_\Phi + sR_a J_1 / c_\Phi + s^2 L_a J_1 / c_\Phi}\ . \tag{9}$$

The maximum difference between the full and the simplified transfer functions is 1.86dB at the knee, see Fig. 5. The simplified expression is simple enough to provide an insight into the system dynamics.

IV. Conclusions

The paper demonstrated an application of the symbolic analysis to mechatronic systems in the SAMD program. As the basic physical laws of the electric, rotational, and translational systems are the same, all the methods developed for the electric domain can be used in the mechanical domain as well.

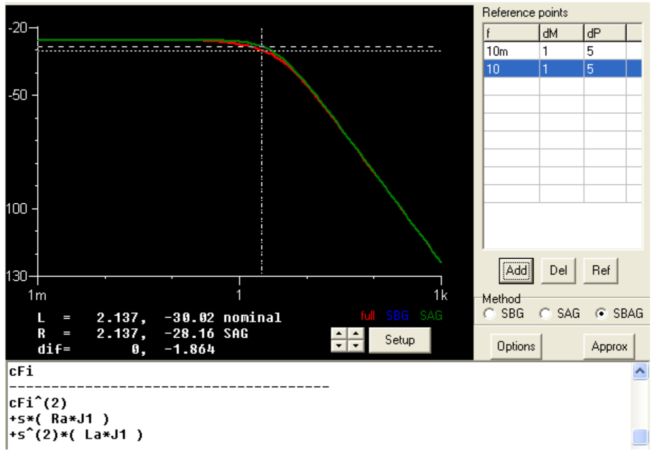


Figure 5. Simplification of transfer function (8) in SAMD.

References

[1] S. Scherber, “Modellierung und Simulation mechatronischer Systeme. Technischer Bericht,” Siemens, Germany, Sie-97-02-001, 1997.

[2] D. Biolek, V. Biolková, J. Dobeš, “(Semi) Symbolic modelling of large linear systems: Pending Issues.” In: Proc. Int. Conf. ISSSE 2001, Tokyo, Japan, 2001, pp. 397-399.

[3] A. Luchetta, S. Manetti, “SAPWIN – A Symbolic Simulator as a Support in Electrical Engineering Education.” IEEE Trans. on Education, Vol. 44, No. 2, May 2001, p. 213.

[4] Z. Kolka, “SNAP – Program for Symbolic Analysis.” Radioengineering, Vol. 8, No. 1, 1999, pp. 23-24.

[5] Z. Kolka, D. Biolek, V Biolková, “Symbolic analysis of linear circuits with modern active elements.” WSEAS Transactions on Electronics, vol. 5, no. 6, 2008, pp. 88-96.

[6] W. Daems, G. Gielen, W. Sansen, “Circuit Simplification for the Symbolic Analysis of Analog Integrated Circuits.” IEEE Trans Comp. Aided Design, vol. 21, no. 4, 2002, pp. 395-407.

[7] J. Kalous, D. Biolek, Z. Kolka, “Project of a Program for Symbolic Analyses of Mechatronic Drives”. In Proc. of IM 2010 Conference, Czech Republic, 2010.

[8] Z. Kolka, V. Biolkova, “Formula Simplification using Two-Graph Method.” In Proc. of IEEE Latin American Symposium on Circuits and Systems, Foz do Iguazu, Brazil, 2010.

[9] Chua, L.O. Nonlinear Circuit Foundations for Nanodevices, Part I: The Four- Element Torus. Proceedings of the IEEE, vol. 91, no. 11, 2003, pp. 1830-1859.

[10] C.W. Ho, A.E. Ruehli, P.A. Brennan, “The Modified Nodal Approach to Network Analysis.” IEEE Trans. CAS, Vol. CAS-22, No. 6, 1975, pp. 504-509.

[11] F.E. Cellier, M. Otter, H. Elmqist, “Bond Graph Modeling of Variable Structure Systems.” Proc. ICBGM'95, Las Vegas, NV, 1995, pp. 49-55.

[12] J. Vlach, K. Singhal, Computer Methods for Circuit Analysis and Design. Van Nostrand Reinhold, New York, 2nd edition, 1994.

[13] W.K. Chen, Active Network Analysis (Advanced Series in Electrical & Computer Engineering- vol. 2). World Scientific Publishing, UK, 1991.

[14] W. Daems, W. Verhaegen, P. Wambacq, G. Gielen, W. Sansen, “Evaluation of error-control strategies for the linear symbolic analysis of analog integrated circuits.” IEEE Transactions on Circuits and Systems-I 46(5), pp. 594-606, May 1999.