Study of a sizing methodology and a Modelica code generator for the bond graph tool MS1

†Audrey Jardin, †Wilfrid Marquis-Favre, †Daniel Thomasset, *Franck Guillemard, $^+$Francis Lorenz

†AMPERE
INSA-Lyon
25, avenue Jean Capelle
F-69621 Villeurbanne Cedex

*PSA Peugeot Citroën
Centre technique de Vélizy
Route de Gisy
F-78943 Vélizy-Villacoublay Cedex

$^+$LorSim
89, rue Jacob-Makoy
B-4000 Liège

firstname.lastname@insa-lyon.fr
franck.guillemard@mpsa.com
francis.lorenz@lorsim.be

Abstract

Complex systems engineering requires new software tools for virtual prototyping which have to be more relevant in order to meet, at the same time, consumer requirements, standardized rules and market law. These have to be more flexible especially concerning file exchange and reusability. Recently the modelling language Modelica seems to fulfill these needs thanks to its concepts of acausality and multi-disciplinary description.

In parallel, the laboratory AMPERE has developed a bond graph-based sizing methodology which, by the use of inverse models, drastically decreases the number of calculus iterations compared to the classical direct approach.

The aim of this paper is to highlight the importance of acausality and structural analysis in a design approach and to study to what extent the proposed sizing methodology can be formulated in Modelica. Then first software implementations of the methodology are illustrated by examples processed by the tool MS1 and its Modelica code generator.

Keywords: code generator; Modelica; MS1; bond graph; acausality; structural analysis; sizing methodology

1 Introduction

Nowadays technological advances have lead to systems which are more and more complex and thus, more and more difficult to design. In the new context of sustainable development, systems have to match ever-increasing pollution standards while engineers have to take into account both higher consumer requirements (like safety, comfort, equipment,...) and financial constraints. In few words, engineers have to conceive faster new safer and cheaper solutions.

One way of doing that is to proceed by simulation which has the benefit to avoid costly manufactures of several impertinent prototypes and then favour gain of time and money.

However virtual prototyping is really efficient only if the engineer is able to accurately model the system, i.e. only if the system is sufficiently described for the given problem. In fact, the hardest tasks of such an approach are:

- finding the good description level;
- being able to express the different physical phenomena implied by this description;
- and representing these in an unified manner even if they involve various physical domains.

For all of these reasons, engineers need a modelling language which:

- allows making connection between all kinds of physical domains.

  → The modelling language has to be multi-domain.
- ensures a sort of continuity at every level of the project cycle. So models have to be usable as well in oriented system softwares during a pre-sizing phase as in more specialized tools in advanced design steps.

  → The modelling language has to be recognized as a standard for model exchange.

- reduces wasted time as much as possible. In fact, it is of the first importance to mutualize modelling efforts which, as mentioned before, are the hardest tasks of such an approach. One way of capitalizing on this is to separate the system description from the design context and thus not to depict the system with *a priori* oriented equations.

  → The modelling language has to be object oriented and to enable acausal description.
- reduces study costs by decreasing dependency towards exclusive software providers.

  → The modelling language has to be a free and non-owner language.

This is just with this in view that the Modelica language and the OpenModelica simulation environment [1] have been proposed. In fact this can explain why, today, Modelica language seems to fulfill a real need for engineers and industrials and seems to present itself as the future standard for model exchange. As a proof of fact, numerous simulation environnements and computer aided design tools like Dymola [2], LMS Imagine.Lab AMESim [3] or Scilab/Scicos [4] can now support Modelica models as well for import as for export.

Starting from this statement, the aim of this paper is to compare some Modelica aspects to the bond graph-based sizing methodology [5]-[12] developed by the laboratory AMPERE[1]. In fact, by using the multi-domain aspect as well as the concept of acausality, it seems legitimate to ask oneself to what extent the proposed methodology can be supported by Modelica language.

The paper is organized as follows. First, section 2 will briefly describe the methodology principles and its benefits compared to a classical design approach. Importance of the acausality concept and the use of a structural analysis will also be highlighted. Section 3 will present one example of the methodology software implementation, the tool MS1 [13], and its newest functionality: a Modelica code generator. Then section 4 will conclude by summarizing the several tackled points and by suggesting future research directions.

2 Bond graph-based sizing methodology towards a Modelica-based sizing methodology?

To understand how some Modelica features can be used or be augmented to support the proposed methodology, it is worth first explaining its main principles. Then importance of an unified and acausal description will prove to be a benefit for carrying out a structural analysis. Finally some reflections will be conducted about the potential of embedding the methodology in Modelica.

2.1 Methodology benefits and principles

Up to now a classical approach adopted by the most of engineering departments consists of a trial and error procedure. For instance consider an actuated load system (Fig. 1) and suppose that the design problem is to find an appropriate actuator so that the load follows a given trajectory (i.e. the hoped-for specification). Once the system has been modelled, the first step of a classical approach consists in:

- selecting more or less arbitrarily an actuator (this depends on the degree of the engineer expertise);
- presupposing the control of this actuator;
- launching a direct calculus in simulation according to these assumptions;
- comparing the calculated load trajectory to the desired specification.

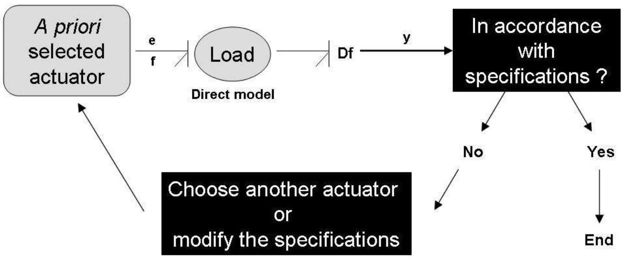

Fig. 1: The classical design approach

However this approach rarely leads to a good solution at the first attempt: it usually requires numerous iterations to find a suitable actuator. This is truer in a technological break context where, by definition, engineers do not have access to any expertise. Moreover this approach can come up to a greater loss of time since:

- in the first case where the *a priori* selected actuator matches the specifications, the engineer has no idea on the margins he has at his dis-

---

[1] Since January 1, 2007, the LAI has merged with the CEGELY and a team of environmental microbiology to become the laboratory AMPERE (UMR CNRS 5005).

posal, and thus whether a smaller and cheaper actuator could be acceptable;

- in the second case where the *a priori* selected actuator does not suit the sizing problem, the result of the simulation does not give any idea on the causes of underdimensioning. The engineer in charge of the study must choose another actuator admittedly more powerful but still more or less arbitrarily.

Finally this iterative procedure can even reveal itself endless as, beforehand, no checking has been made to conclude whether the specifications can be really obtainable by the given structure or not. In that case, most of time, the engineer has to slightly modify the specifications by relaxing some design constraints if he wants to solve his problem.

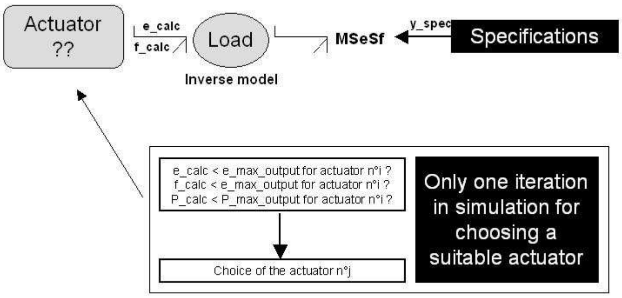

Fig. 2: The laboratory AMPERE design approach for choosing suitable actuators

Faced to all of these drawbacks and strong of its research for 15 years, the laboratory AMPERE has developed an innovative methodology for sizing mechatronic systems ([5]-[8]). Contrary to the classical approach which uses direct model calculus, the key idea here is to exploit inverse models described by bond graph. Considering the same example as before, the main steps of this approach can be summed up into the following points (Fig. 2):

- Step 1: Adequacy

  As explained in more details in section 2.3, this step consists in carrying out a structural analysis. This allows checking if the sizing problem is well-posed and concluding on the possible structural invertibility of the load model (and so on the possibility to inverse the model).

- Step 2: Specification

  Assuming that the load model is structurally invertible, this step consists in establishing the inverse load model corresponding to the given sizing problem (this results in assigning the bicausality on the bond graph model) and simulating it so as to determine variables required at the entrance of the load and that match the specifications[2].

- Step 3: Selection

  As the variables in input of the load are the same as the variables in output of the actuator, the engineer can thus select in a library actuators that appear suitable for the output specifications (e.g. the maximum of the required effort must be inferior to the maximum effort the actuator can supply) (Fig. 2).

- Step 4: Validation

  Finally since actuators have been selected according to criteria only in terms of variables in output, the engineer has to check if these actuators do not overcome their limitations in input (and anywhere else in the inside). This step consists in adding the actuator models to the load model, determining the variables in input by the use of the new corresponding inverse models[3] and comparing the simulation results for these variables to the manufacturer data.

Then these four steps of the methodology are repeated to size each stage of a whole actuating chain (power modulator, energy supplier) and, at the end, to determine the open loop control.

Now that the principles of the methodology have been exposed, it is worth noting some remarks.

First, the methodology does not require any supposition on the actuator control and, by this way, facilitates the engineer study.

Secondly, compared to the classical approach, the inverse methodology drastically decreases the number of calculus iterations. In fact, at the end of the selection step, as the variables needed in output of the actuator are directly determined from the specifications, the engineer is able, after only one calculus, to:

- either eliminate a whole part of the actuator library (whereas each component should have been tested in the direct approach in order to be rejected);

---

[2] One can remark that in this way of calculus, the roles of inputs and outputs are reversed: specified outputs become the inputs of the calculus while the real inputs are the variables to determine.

[3] For the sake of conciseness and clarity, this step has been simplified. More rigorously, another structural analysis must be conducted on the new model including the actuator model to check, in turn, its structural invertibility.

- or decide to manufacture a made-to-measure actuator if none of the off-the-shelf actuators is suitable;
- or slightly modify the specifications if the financial constraints of the project do not allow special manufactures.

In the validation phase, two cases can also happen: either the actuator chosen in step 3 suits the inputs criteria and the actuator is then validated, or the variables needed in input to fulfill the specifications do not correspond to the actuator use restrictions and the engineer must go back to the selection step. As it will be illustrated in section 3.1, in the first case, the engineer can directly conclude that the actuator is relevant for the desired behavior (and this after only two inverse calculations) and can also evaluate the possible oversizing of the actuator. On the contrary, in the second case, the engineer must even choose another actuator but, this time, the comparison between the required variables and the component limitations gives to him the origins of the undersizing (e.g. the actuator does not support such a high supply of power). Thus the engineer must go back to the selection step but with a significant guideline to follow so as to find a suitable actuator.

Thirdly, thanks to the structural analysis, the engineer can check if his problem is well-posed and, if needed, he can readapt, without any numerical calculus, his specifications to be sure that they can be reached by the chosen model structure. Thus the engineer is sure that his approach will succeed in finding a solution.

2.2 Advantage of an acausal description

From a rigorous point of view, a bond graph model initially represents a system in an acausal manner: the equations are oriented only once causality (or bicausality for inverse models) is assigned. Intuitively, the methodology proposed by the laboratory AMPERE can be applied not only for sizing problems but for other engineering contexts too: one only needs to work on the inverse model corresponding to the given problem.

Now, outside the bond graph context, a causal model is only a representation of a calculus sequence (i.e. a set of partially ordered assignments). It thus depends on the study objective and can only be used for this objective. On the contrary an acausal model is only the description of a system (i.e. a set of non-ordered implicit equations), totally independent from what oneself wants to calculate. In this way, the reusability of models described in an acausal form seems to be infinite while the one of causal models reduces itself only to what they are prescribed for.

As Fig. 3 shows, if the engineer chooses a causal approach, he is obliged to formulate one causal model for each problem. On the contrary, if he chooses the acausal approach, the same model can be used for all engineering problems as: analysis, sizing, control design, parametric synthesis, steady state research, ... (Fig. 4).

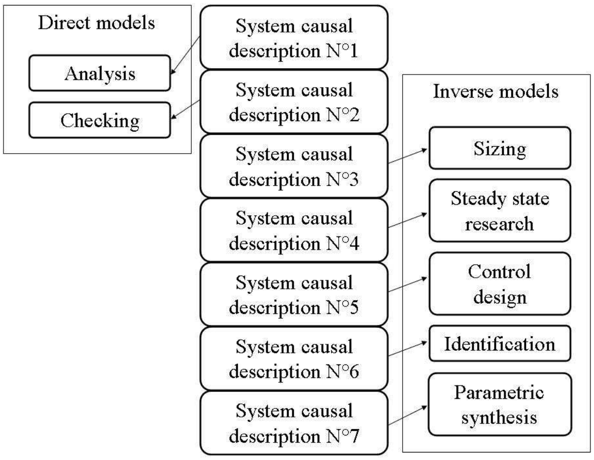

Fig. 3: System causal descriptions required for several engineering objectives

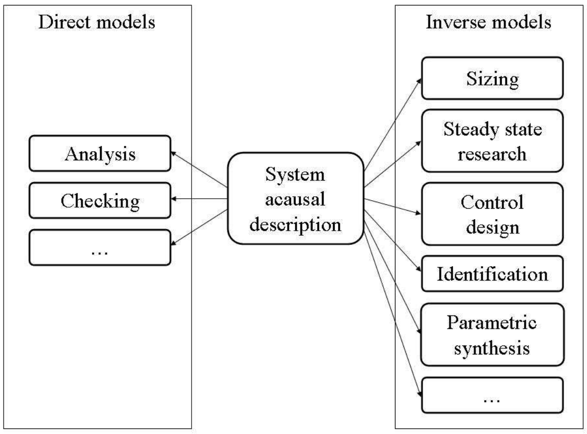

Fig. 4: Only one system acausal description required for several engineering objectives

In practice, this notion of acausality already showed its benefits especially in Modelica language and in bond graph theory.

In the Modelica context, the concept of acausality, added to the concepts of encapsulation and inheritance, enables modelling efforts to be mutualized and librairies to be obtained, librairies that are less redundant (since there is no more need to model the same component in different contexts).

In addition to this, some researches were carried out on how translating different engineering problems into bond graph language. In fact one can remark that each of the proposed procedures starts from the acausal bond graph model so as to construct the direct (respectively inverse) model corresponding to the given problem. To quote just some of them, some works have been done on sizing problem [7][11][12], steady state research [14], parametric synthesis [15], control design [16], characterization [17] and sensibility analysis [18].

2.3 Advantage of a structural analysis

As mentioned before the first step of the AMPERE's methodology involves a structural analysis of the model which the two objectives are checking if the problem is well-posed and verifying the adequacy between the specifications and the chosen model structure.

To understand how these checks can be made, some definitions are introduced and the structural analysis is explained as well as how it can be conducted.

Concerning the concepts [9]:

- a power line is defined as a path for energy transmission between two points of the system (this is an acausal concept);
- a causal path is an ordered sequence of variables connected each one to another by the equations of the system without that a variable appears more than once in the sequence;
- an input/output power line (resp. causal path) is a power line (resp. a causal path) between an input and an output of the system;
- two power lines (resp. causal paths) are said disjoint only if there is no power (resp. no variable) in common;
- when the causality of the whole model has been assigned in order to obtain the maximum number of energy storage phenomena in integral causality, the order of a causal path is defined as the difference between the number of energy storage phenomena in integral causality and the number of those in derivative causality along this causal path.

Given a sizing problem with multiple inputs to determine from multiple specified outputs, checking if the problem is well-posed, in the sense of invertibility, thus consists in finding:

- at least one set of input/output disjoint power lines;
- and, at least one set of input/output disjoint causal paths.

If the required sets exist, then it can be concluded that the model is structurally invertible (i.e. invertible assuming that the equations of the system are locally mathematically invertible): the engineer can thus be sure that his problem is, at this stage, well-posed.

Now, on the contrary, if no set exists, it proves that the model is structurally non invertible. In that case, the procedure stops here until the problem is reformulated. This can be particularly useful for architecture synthesis. In fact if the *a priori* chosen structure does not enable the specifications to be reached, one can imagine another architecture that may satisfy the design constraints. By analyzing the input/output power lines, one can then determine the place an actuator must have in order to control a specified degree of freedom.

Finally once a good structure has been chosen and the model invertibility has been proved, the adequacy, between the specifications and the structure, can be verified. To proceed with this, one needs to check if the time derivability of each specified output is at least equal to the order of the involved input/output causal path. Not only useful for checking, this can then help to write specifications.

2.4 Methodology translation into Modelica language

If previous articles have proved the feasibility of translating a bond graph model into a Modelica model [19]-[22], the key idea here is to study to what extent a bond graph-based sizing methodology can be adapted to Modelica language. If the translation of a bond graph model into a Modelica code can be done *quasi* systematically with the BondLib library [23], the reverse operation is not so easy. Although the concepts of acausality and multi-disciplinary description seem to establish a parallel between the bond graph and the Modelica language, the conversion of a Modelica description into a bond graph model reveals itself like a harder or even impossible task.

In fact if the bond graph is intrinsically bounded to the description of the system energetic structure, nothing imposes to the modeller to depict it into Modelica language. As a proof of fact, a system can be totally described by equations gathered together into the same Modelica class, without any use of Modelica 'connect'. Moreover if 'connect' classes appear in the Modelica code, they do not necessarily represent physical energy exchanges: the Modelica modeller is totally free of choosing his variables for description.

For these reasons, the study of power lines proves to be compromised in a Modelica model and Modelica language does not seem to be suitable for the structural analysis as we have defined it above. However the interesting think of this translation tentative is to highlight that to manage a structural analysis, the engineer has to furnish a minimum set of information about the system and particularly concerning how the different physical phenomena are connected the ones to the others. Besides if we come back to the definitions relative to the structural analysis (section 2.3), one can remark that they can be formulated outside the bond graph context on condition that the concepts of energy storage/dissipative phenomena, power and energy variables be well defined. Thus one can imagine designing a sort of Modelica overlay able to depict the required information of the model.

Actually this way of doing things reveals itself more relevant since the structural analysis does not require the system equations (and so equations described in the Modelica code) but only its energy skeleton. The structural analysis pertains to a step upstream of the Modelica code writing and concerns finally directly the modelling step, where the engineer sets up the system structure and formulates the corresponding problem and specifications. Modelica can then be viewed as a complementary tool to the methodology for model exchange and reusability but not as a tool made for structural analysis.

3 MS1: an example of the methodology software implementation

To illustrate the several concepts previously described and to show how the sizing methodology can be implemented into a program, this section presents the software MS1 with its functionnalities [13]. Two examples processed by it will be used to this objective: the first one concerns the case of a two-link manipulator whereas the second one involves a load actuated by a DC motor.

3.1 Methodology implementation

Structural analysis

One of the MS1 particularities is its module of structural analysis. This functionnality is of course only reserved for the models described into bond graph language since the aforementioned structural analysis requires a minimum information on the system structure. Once the system is modelled into a bond graph representation and once the inputs/outputs of the problem are declared, the software MS1 is able to:

- search all existing input/output power lines;
- search all existing input/output causal paths;
- search all existing sets of disjoint input/output causal paths;
- determine the order of each causal paths or set of causal paths.

So, instead of doing it manually, the modeller can automatically analyze the structural properties of his model. He can conclude on his problem effectiveness and check the adequacy between the results of the structural analysis and his specifications.

Selection/validation step

Another functionnality of the software MS1 is the automation of the selection step. In fact the modeller can define a place-holder for an actuator in his model and, then, the ‘sizing’ functionnality of MS1 enables a sequence of numerical resolution to be automatically conducted. In fact, during this step, MS1 searches in a component library which actuator will be suitable for the given specifications. At the end of the calculus sequence, the engineer has a summary indicating for each actuator:

- its margins compared to what is required;
- and if the component is validated or not.

To illustrate this functionnality, consider the example of a two-link manipulator (Fig. 5). This system consists of a robot made from two solid arms. The first arm is attached to the ground and to the second arm by two pivot joints which are both actuated. This robot is supposed to operate in a horizontal plane and inertias of the actuators as well as the effect of the gravity are neglected.

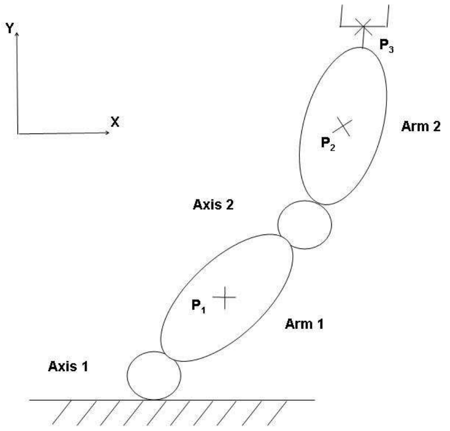

Fig. 5: Two-link manipulator system

Now consider the problem of selecting an appropriate actuating system for the axis 2 so as to the endeffector of this robot follows a given profile in velocity[4]. The selection step, consisting of a research in the electrical drive library of the MS1 database, leads to the following two results:

- a case where the selected component does not suit the specifications (Fig. 6);

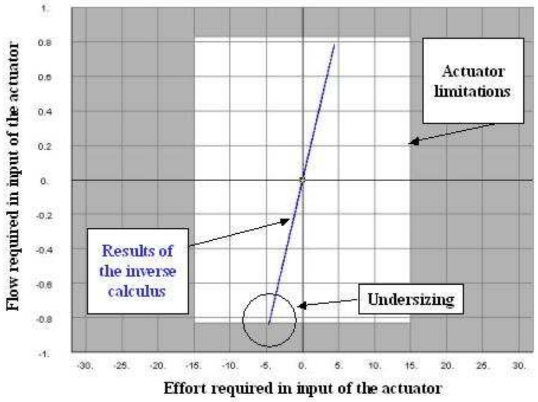

Fig. 6: Validation step: case of an undersized actuator

- a case where the selected component limitations matches with the specified trajectory (Fig. 7).

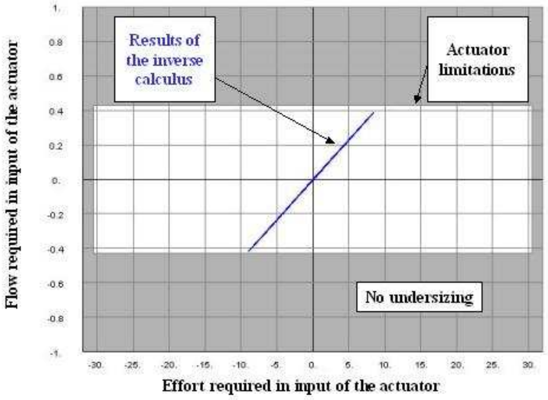

Fig. 7: Validation step: case of a suitable actuator

By representing in an (effort,flow) plane the variables required for reaching the specified output and by superimposing this curve to the manufacturer drive characteristics, one obtains a very convenient way for selecting components and for visualizing causes of under/oversizing. Moreover, as the needed variables are calculated for every instant of the dynamic specification, the engineer is able to detect at which instant the actuator overcomes its limitation and for which duration. Then he can size its component according to the dynamic criteria and, some manufacturer drive characteristics can be taken into account such as the ones for intermittent operation.

3.2 The Modelica code generator: illustration of the acausal description advantage

One of the advantages of the software MS1 is its concept of multi-language platform. Actually, models can be depicted into MS1 in different ways like: bloc diagram, bond graph, NMF network or algorithm. Moreover these models can also be numerically simulated by different solvers: for example, users can lead their numerical resolution by Esacap^{TM} [24], Matlab® [25] or Maple^{TM} [26]. Today one of the newest MS1 functionnalities is its capability to understanding Modelica language. The software MS1 can thus:

- generate automatically Modelica code from any model described into one of the modelling languages previously quoted;
- call for the OpenModelica solver in order to proceed to the numerical resolution.

In fact the generated Modelica code is what is called 'a flat model' in the sense that it only consists of the whole equations gathered into the same class object. Thus neither heritance nor encapsulation are used here. However this model can be interpreted by any existing Modelica compiler and respects, by this way, the wish of the Modelica Association to be proprietary independent.

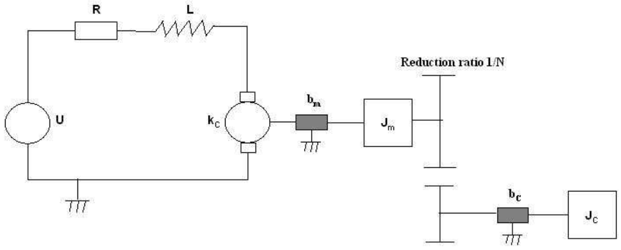

Fig. 8: DC motor actuated load system

The following example will illustrate different Modelica results generated by MS1. Consider a system consisting of a load actuated by a DC motor (Fig. 8) and suppose that the rotor shaft and the load shaft are both infinitely stiff.

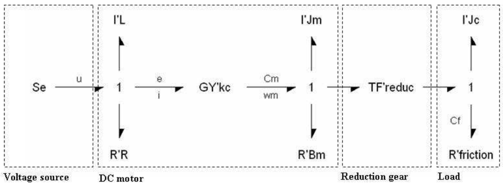

Fig. 9: Acausal bond graph model of a DC motor actuated load system

---

[4] Even if these steps are not explicitly described here, it is assumed that the model is invertible and that the velocity profile is enough time differentiable.

The system is modelled in terms of an acausal bond graph as shown in Fig. 9. In more details:

- the Se-element stands for the voltage source;
- the three I-elements represent the three energy storage phenomena respectively associated to the magnetic energy and the kinetic energies of the rotor and the load respectively;
- the three R-elements enable the dissipative phenomena involved respectively in the electrical circuit, on the shaft and on the load viscous type friction to be described;
- the GY-element depicts the electro-mechanical coupling;
- and the TF-element is associated to the power conserving coupling in the ideal reduction gear.

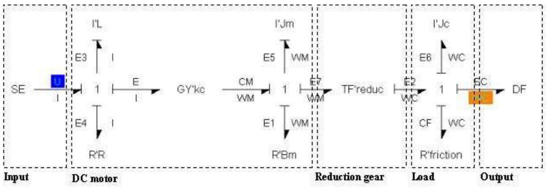

Fig. 10: Causal bond graph model of a DC motor actuated load system

Now consider a first engineering problem which the aim is to analyze the behavior of the load under a given control. Translating this problem into the bond graph language consists only just in starting from the acausal bond graph, defining the effort variable on the MSe-bond as the input, adding a Df-element representing the ideal measure of the load angular velocity and defining the corresponding flow variable as the output. This operation enables to declare which variables are known and which are to be calculated according to the given problem. The causality assignment leads to the bond graph given in Fig. 10 and the Modelica code corresponding to this problem is presented in Fig. 11.

```
class ActuatedLoad
parameter Real
  L = 0.001, R = 8.0, KC = 0.031,
  JM = 1.8E-6, N = 20.0, RC = 0.0001,
  JC = 2.E-4;
parameter Real
  G1 = 0.0, G2 = 0.0;
Real
  EC = 0.0;
Real
  U, I, E4, CM, WM, E1, WC, CF, P3,
  E2, E7, E5, E, E3, E6;
Real
  P1(start = G1), P2(start = G2);
equation
  U = 10*sin(5*time+0.0); ← INPUT
  E6 = der(P3);
  I = P1/L; E4 = I*R; CM = I*KC;
  WM = P2/JM; E1 = WM*1.0;
  WC = WM*(1/N); CF = WC*RC;
  P3 = WC*JC; E2 = EC+CF+E6;
  E7 = E2*(1/N); E5 = CM-(E1+E7);
  E = WM*KC; E3 = U-(E4+E);
  der(P1) = E3; der(P2) = E5;
end ActuatedLoad;
```

Fig. 11: Modelica code associated to an analysis problem for the DC motor actuated load system

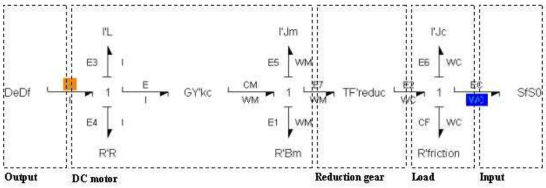

Fig. 12: Bicausal bond graph model of a DC motor actuated load system for open loop control determination

Finally consider a sizing problem where the question is to determine the open loop control of the voltage source so that the load follows a given trajectory. This time the bond graph model corresponding to this inverse problem consists in replacing the MSe-element (resp. the Df-element) by a double detector element (resp. double source element) since the roles of inputs/outputs are here reversed. Assigning bicausality results in the Fig. 12 bond graph model. The corresponding Modelica code is shown in Fig. 13.

```
class ActuatedLoad
parameter Real
  N = 20.0, JM = 1.8E-6, KC = 0.031,
  RC = 0.0001, JC = 2.E-4, L = 0.001,
  R = 8.0;
Real
  EC = 0.0;
Real
  WC, WM, E1, P2, E, CF, P3, E2, E7, CM,
  I, P1, E4, U, E5, E6, E3;
equation
  WC = 0.00193681*sin(5*time); ← INPUT
  WM = WC/(1/N); E1 = WM*1.0;
  P2 = WM*JM; E = WM*KC;
  CF = WC*RC; P3 = WC*JC;
```

```
    E5 = der(P2); E6 = der(P3);
    E3 = der(P1); E2 = EC+CF+E6;
    E7 = E2*(1/N); CM = E1+E5+E7;
    I = CM/KC; P1 = I*L;
    E4 = I*R; U = E3+E4+E;
  end ActuatedLoad;
```

Fig. 13: Modelica code associated to a problem of an open loop control determination for the DC motor actuated load system

One can then observe that both Modelica codes differ only by the equations concerning the input variables (respectively U for the analysis problem and WC for the sizing problem). When the Modelica 'connect' class will be implemented in MS1, we will obtain a model split into four classes (respectively for the dc motor, the load, the input and the output) and only those classes relative to the input and output will change between both problems.

4 Conclusion

Compared to the classical design approach, the sizing methodology, developed by the laboratory AMPERE, offers numerous benefits. In fact with the use of inverse models and structural analysis, this methodology enables the engineer to check if his problem is well-posed and to verify the adequacy of the specifications with his model structure. Moreover this tremendously decreases the number of calculus iterations since it gives, at the selection and validation steps, enough information in order to select another component in the case of an undersized one or to choose an optimal one in the case of several suitable actuators by comparing the margins of sizing.

By emphasing the roles of acausality and multi-domain description, the aim of this paper is to ask if the methodology, originally based on the bond graph tool, may be supported by another modelling language like Modelica. After having proved the importance of acausality and structural analysis in a design approach, it has been concluded that finally the concepts used in the methodology can be defined outside the bond graph context but are not well adapted to Modelica language. In fact the notion of structural analysis requires the description of the system energy structure and thus must be conducted upstream of the Modelica code.

Here the tool MS1 enabled the feasibility of the methodology software implementation to be proved. Functionalities, like the one of automation of the structural analysis or of the component selection in an actuator library, are available. Besides this a Modelica code generator was implemented in order to convert automatically a bond graph model into a Modelica 'flat' model.

In the context of the RNTL-SIMPA2 project, which the aim is to develop a Modelica compiler and integrate it into Scicos and LMS Imagine.Lab AMESim softwares, some researches are currently under progress for designing a module of structural analysis totally independent from any modelling language. Integrated into Scicos and more focused on the GUI, it will rely on the analysis of XML files describing the model structure. Through the GUI, the engineer will thus be guided to formulate his problem in a textual manner, describe his system in terms of energy exchanges and declare which are the known variables and the unknowns of the problem. The engineer will then be able to conduct a structural analysis (and then to apply the methodology) starting from this description, and this without knowing the bond graph theory. Results of the structural analysis will be appear in a textual manner too and a Modelica code of the problem will be eventually created.

Acknowledgement

The authors gratefully acknowledge the partial support of the French National Research Agency (ANR) through the RNTL-SIMPA2/C6E2 project.

References

[1] http://www.modelica.org/

[2] http://www.dynasim.com/index.htm

[3] http://www.lmsintl.com/

[4] http://www.scilab.org/

[5] Ngwompo R.F., Scavarda S., Thomasset D. Physical model-based inversion in control systems design using bond graph representation, Part 1: theory, Proceedings of ImechE Journal of Systems and Control Engineering, 2001, vol. 215, pp. 95-103.

[6] Ngwompo R.F., Scarvada S., Thomasset D. Physical model-based inversion in control systems design using bond graph representation, Part 2: applications, Proceedings of ImechE Journal of Systems and Control Engineering, 2001, vol. 215, pp. 105-112.

[7] Ngwompo R.F., Scavarda S., Thomasset D. Bond graph methodology for the design of an actuating system: application to a two-link manipulator, IEEE International Conference on Simulation, Man and Cybernetics, Orlando, USA, 1997, pp. 2478-2483.

[8] Ngwompo R.F. Contribution au dimensionnement des systèmes sur des critères dynamiques et énergétiques – approche par Bond Graph. Lyon, France: PhD thesis, INSA-Lyon, 1997.

[9] Ngwompo R.F., Bideaux E., Scavarda S. On the role of power lines and causal paths in bond graph-based model inversion, in proceedings of the International Conference on Bond Graph Modeling and Simulation, New Orleans, USA, 2005, pp. 78-85.

[10] Ngwompo R.F., Scavarda S. Dimensioning problems in system design using bicausal bond graphs, Simulation Practice and Theory, 1999, vol. 7, pp. 577-587.

[11] Laffite J., Bideaux E., Scavarda S., Guillemard F., Ebalard M., Moutou C. Modeling in order to size an automotive power train, in proceedings of the 13th European Simulation Symposium, Marseille, France, 2001, pp. 806-812.

[12] Mechin O., Marquis-Favre W., Scavarda S., Ferbach P. A dynamic sizing methodology in the context of an automotive application, in proceedings of the ASME International Mechanical Engineering Congress & Exposition, New Orleans, USA, 2002.

[13] http://www.lorsim.be/Default.htm

[14] Bideaux E., Marquis-Favre W., Scavarda S. Equilibrium set investigation using bicausality, Mathematical and Computer Modeling at Dynamical Systems Special Issue on Bond Graph Modeling, Taylor & Francis, 2006, vol. 12, pp.127-140.

[15] Ngwompo R.F. Développement d'une méthodologie de dimensionnement des éléments d'un système : application au cas d'une suspension hydropneumatique. Lyon, France: Technical report, INSA-Lyon, 1999.

[16] Bideaux E., Smaoui M., Brun X., Thomasset D. Design of a compliant positionning control using an inverse method, in proceedings of the Power Transmission and Motion Control Conference, Bath, UK, 2003, pp. 147-162.

[17] De Giorgi R., Sesmat S., Bideaux E. Using inverse models for determining orifices mass flow rate characteristics, Sixth Japan Symposium on Fluid Power, Tskuba, Japan, 2005.

[18] Borutzky W., Granda J. Determining sensitivities from an incremental true bond graph, in proceedings of the International Conference on Bond Graph Modeling and Simulation, Phoenix, USA, 2001, pp. 3-8.

[19] Borutzky W., Barnard B., Thoma J.U. Describing bond graph models of hydraulic components in Modelica, Mathematics and Computer in Simulation, 2000, vol. 53, pp. 381-387.

[20] Borutzky W. Bond graph modeling from an object oriented modeling point of view, Simulation Practice and Theory, 1999, vol. 7, pp. 439-461.

[21] Broenink J.F. Object-oriented modeling with bond graphs and Modelica, in proceedings of the International Conference on Bond Graph Modeling and Simulation, San Francisco, USA, 1999, vol. 31, pp. 163-168.

[22] Broenink J.F. Bond-graph modeling in Modelica, European Simulation Symposium, Passau, Germany, 1997.

[23] Cellier F.E., Nebot A. The Modelica Bond-Graph Library, in proceedings of the 4th International Modelica Conference, Hamburg, Germany, 2005, pp. 57-65.

[24] http://www.ecs.dtu.dk/esacap.htm

[25] http://www.mathworks.com/

[26] http://www.maplesoft.com/