Transactions on Information and Communications Technologies vol 8, © 1995 WIT Press, www.witpress.com, ISSN 1743-3517

Equation generation in engineering physical-systems problem solving

J.Mª. Díaz de la Cruz Cano, J.J. Scala Estalella
Department of Applied Physics, ETS Ingenieros Industriales, Universidad Politécnica de Madrid, Spain

Abstract

In this contribution we describe the different types of equation generation methodologies that we have encountered in a work of search through the main areas of engineering physics. We elaborate an abstract model of physical system consisting of an oriented graph, and examine these methodologies in relation to the graph topology. We sketch the main features of a formal description language to support the specification of such strategies. This language, described thoroughly in the bibliography, is computable and can be effectively interpreted by a computer program to perform the equation generation automatically when presented to a particular graph.

1 Introduction

Equation generation is perhaps the most important clue to engineering problem solving. Ensuing mathematical reasoning is often viewed as a secondary (throgh necessary) stage for the full problem resolution. While the mathematical reasoning depends mainly on the available knowledge and practice of the engineer, equation generation is often a *creative* activity. Of course, it depends on previous practice and knowledge, but there is allways something new, not previously encountered, that makes every problem *original*. We can say, drawing from general agreement, that equation generation is the most genuine activity of engineering problem solving. Academic problems represent a broad and available source suitable for our study.

Academic problems are generally grouped in collections stored in books. Every collection has a certain set of physic *laws* and of equation generation *rules*. A problem collection on, for instance, electric circuits, has as physic laws

the Ohm law, voltage addition and current equality in serial combinations, current addition and voltage equality in paralell combination, and current-voltage relating laws for capacitors and autoinductances. Exhaustive application of former laws may constitute a rule for the equation generation. However, other *strategies* leading to easier mathematical resolution have been described. The branches or the nodes rules simplifies the ensuing mathematical treatment of the problem. Analogue situations are encountered in other physics domains such as bar structures, magnetic and thermodynamic circuits etc.

We would like to emphasize the application of our work. According to our appretiation the engineering comunity has plenty of system simulation programs for its professional work. These programs must be fed with the sets of mathematical laws that govern the behaviour of each constitutional block as well as initial condition values when dealing with dynamical simulations. In order to do it, the engineer must have digested the system he wants to simulate. Our work, however, performs the digestion for him and as a result it produces the sets of equations. These sets of equations can be used in two fashions. First, the engineer may want to examine the equations to notice the type of mathematical problem involved and decide on the methodology, software or any other resources to use. Second, he can use these equations as input to other mathematical simulation software.

Experience may induce strategies for producing a simpler set of equations when a professional engineer works in a particular problem. This pretious knowledge can be writen in form of rules as described in this contribution. Then a computer system will automatically apply these rules, thus simplifying a possibly more complex problem.

In our reserch group we have made a wide catalogue of equation generation methodologies. We found that, even for the same physics scope, and so, for the same physical laws, different collections used different rules leading to different (though equivalent) sets of equations. Our main concern was to feature the equation generation process in as many as possible domains to lay down the basic requirements of a formal description language. This language was part of an overall project whose sake was the production of an automatic generator of wide solvable problem collections in the area of technological Physics.

In this paper we present the results of our work. We begin by formalizing the main concepts involved as a necessary step before the ensuing computational approach.

2 Physical systems as directed graphs

In order to work properly with the concepts mentioned in the previous paragraphs, we must find sound mathematical counterparts for them. If we take a look at the varied physical-sketches of *figure 1*, we infer some general features of problem systems. They are built up from a finite set of elementary systems that interact through a definite group of mutual relations.

The first skeme represents an electric circuit. It may be conceived as a set

of elementary electrical branches (R,L,C,U) put in a serial array. The second skeme represents a cinematic device built up from a couple of bars (B1,B2), a fixed cylinder (C1), a rolling cylinder (C2), a fixed point (O1) and a slider (G). Bar B1 is articulated to O1 and the center of C2. Bar B2 is articulated to slider G and to bar B1, etc. The third skeme represents a thermodynamic circuit comprising a pump (B1), a turbine (T1), a boiler (C1) and a cooler (E1), arranged serially. Finaly, skeme 4 represents a bar structure array made up from a definite set of bars that articulate mutually in their common knots.

Each of the elementary systems mentioned above has a set of proper variables. Electrical branches have a proper voltage and a proper current. Each bar has a center-of-mass position, a length, an orienting angle, an angular velocity, etc. Hence these variables will be referered to as *proper variables* of the elementary systems.

We must add some commentary to the precedent paragraphs. Sometimes, interactions also have their own proper variables. A rolling relation beetween a wheel and a plane define a point-of-roll and a velocity of displacement for that point. Two elementary branches arranged parallely define a current and a voltage of the new branch formed that could be conceived as the current and voltage of the paralell interaction, etc. If we draw farther in this direction, we can see that the difference beetween an elementary system an an elementary interaction becomes more and more subtle. In the following lines we shall treat them uniformly. Each elemntary system and interaction are particular instances of a larger class of general elemntary systems and interactions.

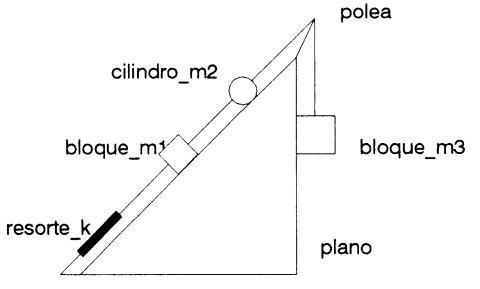

figure 2a

It is not difficult to think of a problem collection as a large sequence of topologic variations of the previous skemes followed by particular specifications for a subset of the union of the proper variables of all elementary systems and interactions, leaving the rest as unknowns. We shall formilize the previous ideas in the following set of definitions.

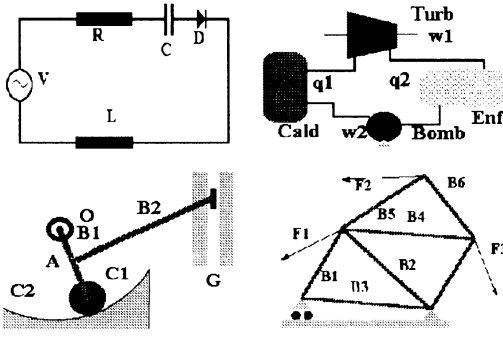

figure 1: physical systems with underlying similitude

Definition 1. An elementary system class C is a n-uple $(q_1,..,q_n)$ of proper variables with a definite

set of relations $R=\{R_1(q_1,..,q_n),R_2(q_1,..,q_n),..,R_k(q_1,..,q_n)\}$ among them. We can specify $C((q_1,..,q_n),R)$ for a particular elementary system. □

Definition 2. A physical-system $\mathfrak{G}$ is a directed graph whose vertices belong to a definite elementary system class. So, we can formalize a physical system as a triplet (N,B,S,f,g,h), where N is a finite set of vertices, B is a definite set of branches, S is a set of elementary system classes, f ang g are functions assigning each branch its initial and final vertices respectively and h is a function assigning each vertex an elementary system class of S. □

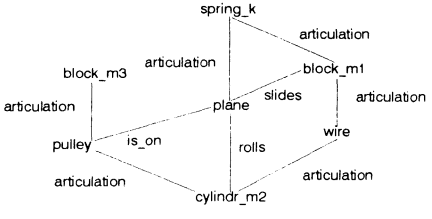

figure 2b

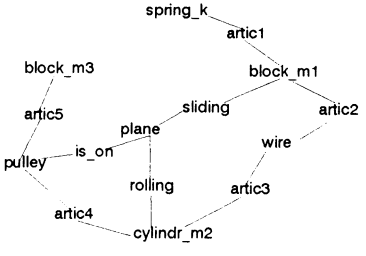

figure 2c

We could think that a more natural approach would have been to separete systems from interactions an let the latter be represented by the branches of the graph, as can be seen in *figure 2b*. The figure shows the graph representing the physical system of *figure 2a*. Elementary systems form the set of nodes of the graph while teir interactions determine its branches. The graph defines completely the physical system. However, a slight modification on it results in *figure 2c* where systems and interaction are treated uniformly. We can now formulate in precise terms all the knowledge that we believe is relevant to determine equation generation laws, strategies and rules as we pointed out in the preceeding epigraph. We shall bring these ideas into further formulation in the next epigraph.

3 Equation generation steps

The equation generation for a given physical system $\mathfrak{G}$ can be conceived as the translation of the relations linking the elementary systems $S_1,..,S_n$ in $\mathfrak{G}$ into relations constraining the values of the variables of $S_1,..,S_n$. Any interaction I,

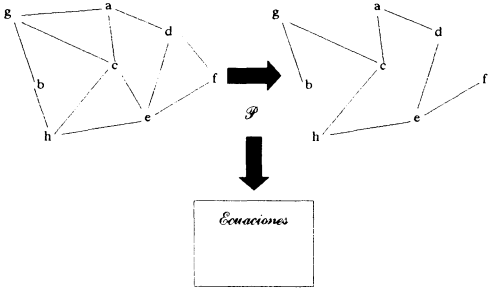

figure 3

represented as a node of $\mathfrak{C}$ connected to other (generally two) S_i,S_j nodes will determine one or more relations among their respective variables.

We can think of a branch-reduction process consisting of a progresive graph prune process isolating the component nodes of the graph as equations fall as ripe fruit. *Figure 3* represents this process.

Besides the equations generated from system topology, there are equations linking the variables of each elementary system. An electric resistance verifies the equation u=Ri independently of how it is connected in a given circuit. So, we can identify two sources and so, two steps in equation generation: pertenence to an elementary system class and topology of the overall system.

Each elementary systemam class, as defined in the previous epigraph is defined by an own set of relations among its variables. So, it is just a matter of *particularization* to reproduce them for every elementary system belonging to a particular system class. The most interesting point is to formulate the strategy that rules the generation of the second type of equations. We have counted three types of strategies: blind, selective and iterative.

Blind strategies take a branch connecting two nodes and generate one or more equations. Iterative strategies need some kind of look-around before producing an equation. Selective strategies act according to a plan. There is a certain schedule based on the topological information of a system, that is, on the way nodes are connected in the graph. We shall treat them in the following epigraphs.

4 Blind generation

This strategy is based on the single principle: one branch -> one or more equations. The following rule pattern can formalize what kind of reasoning we are refering to:

$$\text{If } (\text{branch}(n_1,n_2)\in\mathcal{G})\text{And}(\text{branch}(n_3,n_4)\in\mathcal{G})\text{And}(h(n_1)=C_1)\text{And..And}(h(n_4)=C_4)) \text{ then}$$
$$(R_1(n_1.q_1,..,n_1.q_k,n_2.q_1,..,n_2.q_m)\text{And .... And } R_n(n_1.q_1,..,n_1.q_k,n_2.q_1,..,n_2.q_m))$$

where $\mathcal{G}$ is the graph formed by the physical system $\mathfrak{C}$. For instance, the rule

$$\text{If } (\text{branch}(n_1,n_2)\in\mathcal{G})\text{And}(\text{branch}(n_2,n_3)\in\mathcal{G})\text{And}(h(n_1)=\text{Any})\text{And}$$
$$(h(n_2)=\text{Paralelo})\text{And}()h(n_3)=\text{Any})$$
$$\text{then}$$
$$((n_1.u=n_3.u)\text{ And}$$
$$(n_2.i=n_1.i + n_3.i))$$

generates the equality of voltages in two paralel circuit elements and the additivity of currents of the resulting element (that is represented by the interaction which produced it). Once a set of branches has been mentioned by a successful rule, it is removed from the graph, so other rules may apply until producing a totally disconnected graph, when the process ends.

This kind of rules can be easily handled by many rule-based expert-systems production tools (Guru, Kappa among the most popular PC available ones) . The preceeding schema has limited applicability. It is equivalent to that used in [2].

5 Iterative generation

The preceeding methodology fails to cope for problems where a lot of interactions sum their actions on some particular system. For instance, let's take a lot of bars $B_1,..,B_n$ articulated at the center of a cylinder C_1 and let $A_1,..,A_n$ be the graph nodes representing the consequent interactions. We should say

$$\sum (A_i.\vec{F}) = (C_1.m)\ (C_1.\vec{a})$$

But it can't be represented by rules of the preceeding format. There is still another source of trouble. If we want to formulate an equation generation strategy valid in all the problems of a certain collection, we must formulate it independently of any particular system $\mathfrak{S}$. So we shouldn't know in advance how many bars are articulated to C_1. We don't even know if there will be any C_1 or any bar at all. Something is evident we have to search through the graph for articulations A_i connected to cylinders C_j. Group all found articulations and then apply the previous rule. The iterative operation need not be an addition. It could be a multiplication or any other operation as well. Presently, we don't know of any available expert systems shell where we could formulate and apply properly rules of the previous kind. So we have defined a system to cope with this kind of equation generation. The general rule paradigm can be formulated as follows:

FOR ALL n IF n BELONGS TO CLASS C_i THEN
  $n.q_l$ = ITERATE
    OVER $n_1,..,n_n$
    BELONGING TO G
    SUCH THAT $R(n,n_1,..,n_n)$
    THE EXPRESSION $f(n.q_j,n_1.q_{11},..,n_n.qn_m,f_{k-1})$
    STARTING WITH f_0
  END ITERATE

For instance, take a mechanical problem like that of figure 4a.

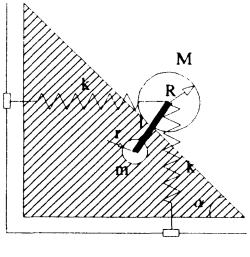

figure 4a

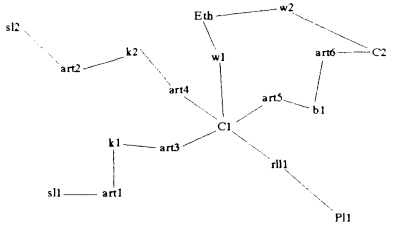

figure 4b

We can see a physical system built up from two elastic springs k_1,k_2, two cylinders C_1,C_2, two sliders sl_1,sl_2 and a plane pl_1. Mutual interactions are articulations (art_1,..,art_6) and a roll (rll_1). Each of these interactions acts as a link between two of the previous elementary systems. Figure 4b represents the graph determined by the global system. Equation generation on this graph stems from elementary class pertenence, blind generation and iterative generation. Relations between forces and accelerations for cylinders belong to the first type of generation. Equalities relating positions of articulation points and centers of cylinders belong to the second type and can be captured by rules of the type described in the previous section. Additivity of all forces acting on a cylinder belongs to the iterative type and can be captured by the following rule. Figure 4c represents schematically the latter two technics.

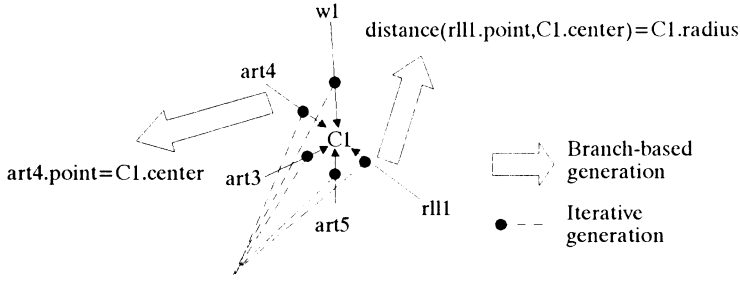

figure 4c

```
FOR ALL n IF n BELONGS TO CLASS cylinder THEN
  n.F_x =          ITERATE
                   OVER n_1
                   BELONGING TO G
                   SUCH THAT (h(n_1)=Art)And(branch(n,n_1))
                   THE EXPRESSION +(n_1.F_x,f_k-1)
                   STARTING WITH 0
                 END ITERATE
                 -
                 ITERATE
                   OVER n_1
                   BELONGING TO G
                   SUCH THAT (h(n_1)=Art)And(branch(n_1,n))
                   THE EXPRESSION +(n_1.F_x,f_k-1)
                   STARTING WITH 0
                 END ITERATE
```

We have analysed the possibility of computer implementation of an expert system shell capable of dealing with rules of this kind. We have built one that has proved efficiently in a variety of domains. Especially in those where elementary systems interact with a non-predetermined number of other systems.

The complete definition of this type of rules requires the specification of the types of conditions we can formulate in the SUCH THAT field. It would be out of place to enumerate the complete specification language, but we can say that allows for basic graph relatons, class pertenence and logic combination through connectives and quantifiers.

This type of rules govern equation generation in free-body diagram technics [3]. In fact, the underlying phylosophy is to concentrate on each elementary system belonging to a definite system class, then list all interactions affecting it and add their effects in the variables of the elementary system. It is more general than the bond-graph technique described in [4].

6 Selective generation

The previous type of rules doesn't allow for system architecture based strategic planning. They produce exhaustive equation generation. All rules are applied whenever they can be applied. Some times we have a redundant set of rules so that exhaustive application may lead to overgeneration. However, this excessive number of rules may allow us to select which ones should be applied in order to simplify the resulting mathematical problem. This kinds of strategies are very familiar to students in areas such as electric circuits, bar structures, linear programming, etc. These strategies are allways formulated - or can be easily translated to - graph theory expressions (elementary circuits of a spanning tree in a circuit graph, etc). So, we have developped a language with sufficient expressive power to bear all possible formulations of equation generation strategies based on the graph defining a physic system. Incidentally, this language happens to bear the previous types of strategies that could be considered as particular cases of this one. It is based on a first order logical language with user-definible primitive symbols an a set of built-in graph relations. It also allows for the inclusion of rules of the iterative type. SUCH THAT conditions are filled with first-order formulae whose free variables contain those of the OVER field. THE EXPRESSION field is filled with a term of the language containing the free variables just mentioned. Iterative constructs are terms of the language whose variables are those of THE EXPRESSION field minus $n_1,..,n_n$. A complete specification can be found in [1].

The elementary circuits ecuation generation technique can be formulated in the expressive construction:

$\forall x,y$ SpanningTree(x,G) And ElementaryCirc(y,x,G) $\Rightarrow$
0= ITERATE
OVER n_1
BELONGING TO G
SUCH THAT CircuitElement(n_1)
THE EXPRESSION +(n_1.voltage,f_{k-1})
STARTING WITH 0
END ITERATE

The main feature of these strategies is that they choose an *adequate* of

traversing the nodes and branches of a graph to generate equations *according to a plan.* The generation proceeds *on shedule.* So, we can design strategies for avoiding equation coupling, non linearities, etc. We have designed several strategies in different domains and even multiple ones for the same problem type. It has given us a formal and sound tool for diccussing, comparing an evaluating different approachs to quation generation in various areas. We found that the chart of *figure 5a* reflects approximately the share of each methodology in university physics in our university. However the rule type distribution in selective generation obeys (also estimatively) the pie figure. So, although common academic systems require iterative or selective type rules, the bulk work seems to lay on blind-type rules. These brief results complete the introductory analysis we wanted to expose in this paper.

8 Conclusion

We have exposed the summary of the work we have developped in this area of research. A complete specification of the grammar and computer system will be sent on request to any member of the engineering community interested in it. We have described the three basic types of methodologies that we have found in different areas. This is part of the work developped in [1] aimed at featuring engineering problems in order to formalize their main characteristics for a later computational treatment. This computational treatment results in the automatic generation of equation sets of problems of a given type. Equation generation can be formalized in sets of rules with precise meaning. This meaning can be interpreted by a computer system to carry out the generation when faced to a variety of systems described as graphs according to the conventions above.

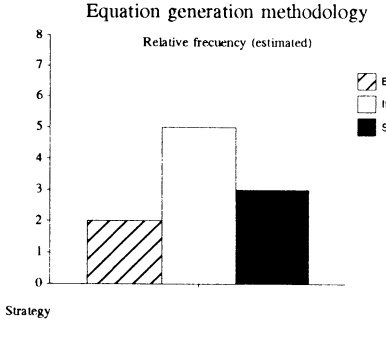

figure 5a

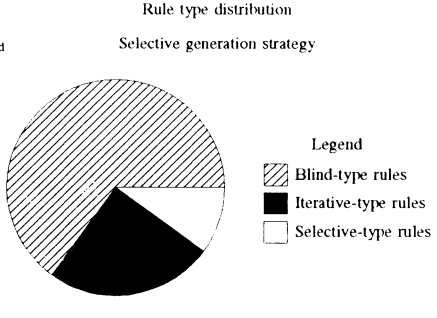

figure 5b

It would be a good exercise to have a collection of generation strategies from different domains and different people stored in a *strategies bank* available to the whole community.

References

1. Díaz de la Cruz Cano, J.Mª. "Generación y solución automática de problemas físico-tecnológicos".Doctoral thesis. Madrid. April, 1994.
2. Giroire Brousse, H. "Un système a base de connaissances pour la generation d'exercises dans des domaines liés au monde reel". Doctoral thesis. Paris. 1989.
3. Van Heuvelen, A. "Learning to think like a physicist: a review of research-based instructional strategies". American Journal of physics. October 1991.
4. Blundell, A. "Bond graphs for modeling engineering systems". Ellis Horword. 1982.