Development of a Computer-Aided Design Tool for Control Engineering with Protégé

Isaías García[1], Carmen Benavides[1], José Ramón Villar[2], Francisco J. Rodríguez[1], Francisco Javier Panizo[1], Ángel Alonso[1]
[1]*[isaias.garcia;carmen.benavides;diefrs;diefpg;dieaaa]@unileon.es*
Control Engineering and Systems Group, University of León (Spain)
[2]*villarjose@uniovi.es*
[2]Computer Science and Artificial Intelligence Group, University of Oviedo (Spain)

Extended abstract

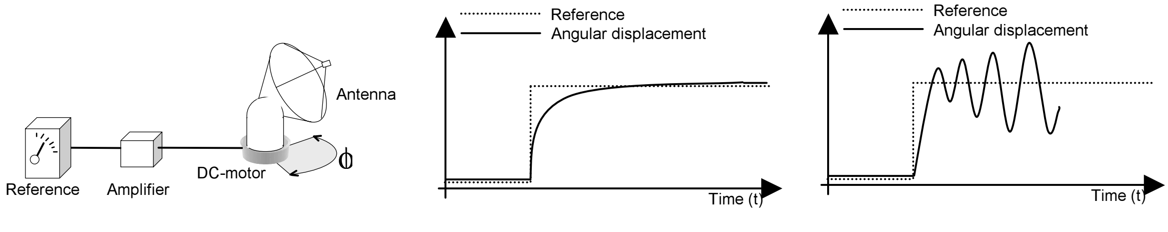
The traditional use of computers in engineering is based on their capability of performing fast numerical calculi. Regarding Control Engineering, the computer has been extensively used for representing and managing models of the systems being studied, performing control actions or running applications like the so called Computer Aided Design (CAD) or Supervisory Control and Data Acquisition (SCADA) tools. All these software are based on traditional representation of data, using approaches from structured languages or object-oriented approaches.

As well as using traditional software engineering techniques for developing software, some other techniques coming from the field of Knowledge Engineering have emerged and been successfully applied to fields mostly related to the so called information technologies. These techniques emphasize the explicit conceptualization of the knowledge of the domain as well as the reusability of the structures where this conceptualization is represented among different domains and applications.

Our work tries to extend the Knowledge Engineering techniques to the building of engineering software and in particular to the control engineering software applications. We are more interested in the advantages of having an explicit conceptualization of the engineering knowledge rather than in the reusability issue (at least for the moment). Being able to have a flexible representation of the domain of control engineering would result in conceptually enriched software that could help when working with this field of knowledge.

The concept “control” is a very general one. It can be applied to very different areas, from the living organisms to the man-made artefacts. In general, control may be regarded like the mechanism that allows a system (biological, electrical, mechanical, etc) to exist in time and to respond to the variations in its environment in such a way that allows it to keep existing. One could think for example in the temperature regulation system of the living bodies that allows them to keep a constant internal temperature despite the variations of the external ones. In the case of man-made artefacts control would permit those artefacts to work in a proper manner (without breaking down, for example) despite the variations in external conditions. We can think of an antenna positioning system, for example, where the desired behaviour would be reaching the orientation (angular position) according to the reference introduced.

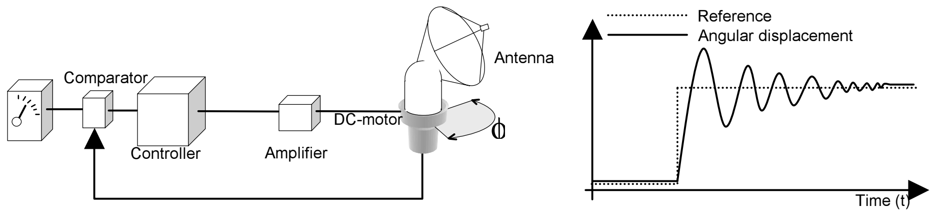
A given system, like the one of the antenna previously presented, could properly function with a schema like the one in the figure 1 (a), where a reference is established by a human or a machine. This reference, duly amplified is fed into a DC motor, which in turn is the one that produces the desired angular displacement in the antenna. If the system works without any kind of problem for any variation in the reference it is said to be stable without the need of any kind of control, as is represented in figure 1 (b). But it is more probable the situation in which, for some variation in the reference, the system will become unstable, as is represented in figure 1 (c). In this case, some control action must be taken.



(a) (b) (c)

Figure 1

The typical control schema consists of building a feedback loop where the controlled variable (the angular displacement in the example) is measured and compared with the reference. The result of this comparison (the error) is used to measure the deviation of the actual displacement compared to the desired. This signal is feeded into a device called controller (usually an electronic circuit) that, properly designed, allows the whole system to become stable. This situation is presented in figure 2 (a) and (b).

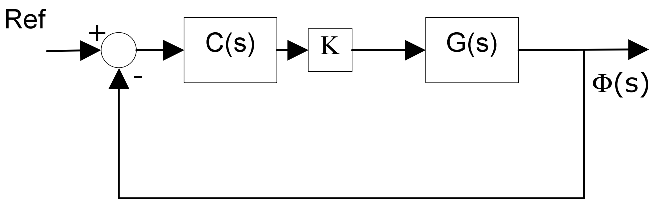


(a) (b)

Figure 2

The central task in control engineering is finding the values of the elements that build up the controller block. Those elements, in the case of an electronic circuit would be values of resistances and capacitors, as well as their correct configuration. The process of building such control schema is iterative: First, an initial design is obtained based on the required specifications, then, this design is tested (analysed) in order to validate it. If the given system doesn't meet the performance required then a new design stage is accomplished, and so on until the analysis eventually fits the specified requirements.

In order to being able to manage the systems involved in the control engineering domain, one must obtain some kind of representation (some model) to work with. The first assumption is the lumped parameters model, than concentrates the behaviour of the system into isolated "ideal" components like resistors, capacitors, inductances, etc. Then, a mathematical model is built by means of ordinary differential equations (ODE) that describe the dynamics of the system. In traditional control theory, the next step is to apply a transformation (the Laplace transformation) to these equations of time that converts them into algebraic equations which are much more easy to deal with. Each of the functional elements in the system is represented by a quotient of polynomials (called the transfer function) that represent the relation between the output and the input signals of the given element. Thus, the system can also be represented by means of a block diagram like the one in figure 3.



$$G(s) = \frac{(s+1.7)}{(s+2)(s-3.2)}$$

Figure 3

While the transfer function of the system to be controlled (G(s)) is known (from the equations defined in its model), the design problem consists on finding the coefficients of the polynomials that build the controller's transfer function (C(s)). The reasoning regarding the election of the parameters of the controller is usually done at a high level of abstraction. For example, one of the design methods (root locus method) deals with the graphical representation of the roots of the polynomials of the transfer functions and other mathematical expressions that can be calculated from these transfer functions.

We could conclude from this short (and simplified) description that the design process "uses names that stand for the graphical representation of some mathematical property of a transformed version of the differential equations of the simplified (lumped parameters) model of the real system". One can see that there are many levels of abstraction up to the one where control experts make their reasoning in the design process. While this is convenient from the point of view of simplicity, the connection and the relations among the different levels of representation is lost, what causes many problems in different situations: One may think about the manufacturing process, where some people puts performance specifications to the real system while the control engineer deals with those high-level concepts, or about the learning process, where the students learn how to manage the roots of the polynomials but aren't aware of what is the physical meaning that lies under this manipulation.

Building software systems that could take into account these different levels of abstraction would be a great advance in order to improve all the communication processes involved in the design of a control system (from the integration of manufacturing systems, better computer aided design tools, intelligent tutoring systems, etc).

In order to be able to explicitly represent all these abstraction levels in such a complex domain, the techniques from Knowledge Engineering could be the suitable ones. To test the validity of the approach, an application is to be built, in particular a knowledge-based system for the aid in the design of a (lead-lag) compensator for a linear system (this is a well known problem in the field of control engineering). The aim is to represent the concepts and the steps involved in the process of designing such compensator in such a way that the system is able of giving some kind of explanation at different levels of abstraction.

An ontology of the domain is being developed, finding the concepts, relations and tasks involved in the process at hand. About the concepts, the basic one is the system, which represents any entity with a boundary, able to receive signals and exerting actions from and into its environment. The system, so defined, may be composed of and so subdivided into subsystems up to a point where it cannot be further subdivided. This conceptual framework leads to the need of representing the compositional relationships which may be described at different levels (lumped parameters representation, block diagrams, etc). Because of the mathematical nature of the representation of the models, mathematical concepts must also be represented. There are "neutral" mathematical concepts like polynomial, coefficient, expression, root,… and "domain specific" mathematical concepts (they are actually some kind of alias or abstraction of the neutral mathematical concepts) like pole, zero,… In order to being able to give some kind of explanation, the causal relations must be taken also into account

About the ontology of tasks, it has been divided into different levels. At the bottom, basic tasks explicitly represent the expert's design rules; at higher levels, the tasks are grouped into coherent groups. Finally, a higher level of tasks implement the control strategy of the reasoning process. Instead of using some kind of design problem solving method, an agenda-like control mechanism with some variations has been chosen because one of the aims of the project is that a control engineer is able to build his/her own structure of tasks and this agenda mechanism is similar to the one used in the control design processes.

Protégé has been chosen as the tool to build the ontology and the knowledge base against which the application will be run. This tool has evolved over a long period of time and now is a mature tool with an exponentially increasing number of users. It has become the de-facto standard tool for representing semantic information and has many possibilities for implementing different reasoning strategies on the data. Another advantage is the ease of programmatically using the knowledge base by means of a complete Java API, which allows the implementation into code of the task structure previously outlined.

The problems and results found so far in this project will be presented, with special emphasis in some difficulties in the conceptualization of specific characteristics regarding control engineering. Also, the link between the Protégé knowledge base and the application will be described, including the link between the structure of tasks and the code used to implement them.