Solving Exercises on Optimal Control by the Conjugated Gradient Method: Benefits of Using a Remote Approach Based on the Internet

Benedito Renê Fischer [1]*,Juarez Bento da Silva* [2]*, João Bosco da Mota Alves* [3]*, Gustavo Ribeiro Alves* [4]

1 São Paulo State University – UNESP, SP,Brazil, brfischer@rc.unesp.br
2 Southern University of Santa Catarina, Araranguá, SC, Brazil, juarez@unisul.br
3 Federal University of Santa Catarina, Florianópolis, SC, Brazil, jbosco@inf.ufsc.br
4 Polytechnic Institute of Porto, School of Engineering, LABORIS, Porto, Portugal, gca@isep.ipp.pt

Abstract: A remote application of a computational package for solving optimal control problems by the conjugated gradient method is implemented by the use of Internet technology. The implementation liberates the user from the details of the programming language, allowing him to focus in the Optimal Control problems. The implementation also provides a manner to protect the source code of the computational systems.

Keywords: Conjugated Gradient Method, Optimal Control, Internet Technology

1. The Conjugated Gradient Method

The Conjugated Gradient Method (CGM) was originally proposed by Hestnes and Stiefel [1] in 1952, for solving a set of linear and simultaneous equations in *n* steps. On its general form it may only be used for optimizing functions in the Rn domain. By the end of the 60's the method was however extended to the solution of optimal control problems, in the case involving the functional optimization in the Hilbert domain, more general than the Rn domain. One of the advantages associated with this method is its application to optimal control, in systems described by non-linear state equations with also non-linear cost functions. The method may even be applied to systems with saturation restrictions, as described by Fischer in [2].

Although the MGC is a purely numeric method, it requires less memory than other alternative methods based on dynamic programming styles, which may be regarded as one important advantage. Generally speaking, numerical problems in engineering were solved using the FORTRAN programming language. The more user-friendly programs usually included a set of routines, to which the user was supposed to add another set of routines describing the dynamics and cost functions of the system under analysis. Fischer describes in [2] a package containing a set of routines that provide the optimal control for four different types of control inputs belonging to the following Hilbert domains: continuous functions, set of parameters, sampled inputs, and pulse width modulation. The described system also accepts restrictions imposed on the state and control variables. As the system solves the minimization problem through the Conjugated Gradient method, it was appelidated of GRAMIN (GRAdient MINimization).

The major disadvantage of such a computacional system is that it requires the user to develop his/her own set of routines describing the the state equations, the adjunct equations, and the cost function, and then compile all his/her routines together with the routines that are part of the computacional package. This implies the user access to the source code of the computacional package and hence to all the programming details and internal structure. This scenario impairs any source code protection and difficults the user from focusing on just the optimal control problem, as he/she has to deal with the details of the programming language and sequence. It would thus be beneficial to have a system where the user could directly enter the expressions defining the state equations, the adjunct equations, the cost function, and the expression for the cost function gradient. This is exactly the situation where Internet-based technologies may allow developing efficient and more user-friendly applications implementing the GRAMIN program, while also protecting the original source code.

2. A remote implementation of the GRAMIN application

As originally implemented in FORTRAN, GRAMIN is divided in two parts that must be compiled together. The 1st is the core program where the (a) minimization algorithms based on the conjugated gradient, (b) the Runge-Kutta and Simpson integrals solving routines, (c) the internal product routines, and (d) the program flow control, are coded. In sum, this is the independent part of the application. The 2nd is the dependent one that must be defined by the user, in the form of routines, and that consists of the state and adjunct equations, and the cost function, for the system for which the user is trying to obtain the optimal control.

In our approach, we decided to implement in a server the independent part of the GRAMIN application, using an Internet-based technology. The dependent part may be remotely specified by the user, by filling in a special form and then, at the end of the process, submit it, which causes the data to be sent over to the server where the independent part is running. This solution frees the user from knowing the programming details of the implemented application and how to implement routines in FORTRAN (for specifying his own information), with the additional advantage of protecting the source code of the GRAMIN application, as the user will no longer need to have access to it.

We started by using the Delphi programming language, together with ASPs and CGIs (which allowed the user to send a part of GRAMIN application to the IIS – Internet Information Server), for implementing the proposed solution at the server side. For the client side we developed a simple HTML page enabling the user to fill in a form with the application-dependent GRAMIN functions. This page was accessible through any simple web browser. On a later approach, we used C together with PHP and the Apache technology implemented at the server side. However, both approaches did not eliminate the need to interpret or even compile the application-dependent functions. In other words, it was necessary to have a sintax and semantic checker analysing the forms submitted by the user. This problem was only solved by using the JavaScript language, as it supports a function, named **eval()**, which evaluates strings, provided the code is syntactic correct. We surveyed current literature to check if this function was also available within the original Java language, and got to the final conclusion that it is only supported in JavaScript. This language can either be executed at the server or the client sides, depending on how the call to the script is defined.

In conclusion, the JavaScript language and the eval() function eliminated the need to interpret or compile the forms submitted by the user. The only negative side of this solution is that the JavaScript language is totally interpreted, and thus coding the entire GRAMIN application in JavaScript could lead to relatively long computational times. With the processing power and memory capacities of current machines this potential problem is no longer significative.

On its current implementation, the GRAMIN application has been translated into JavaScript, residing at the server side, together with an IIS web server. At the client side, there is an HTML page that may also establish the initial call to the GRAMIN application.

To display the results returned by the application, at the client side, we decided to develop a **Java applet**, which is initially dowloaded to the client machine. This applet contains all the graphical functions that allow the user to see the results returned by the GRAMIN application.

3. Results

With the implemented solution, an user may solve optimal control problems, by the conjugated gradient method, without needing a conventional programming language and a graphical package for displaying the results. A web browser supporting both the Java and the JavaScript languages will suffice. The application accepts control inputs within the following domains: continuous functions, a set of parameters, sampled inputs, pulse width modulation, and a generic user-defined domain.

The following figures illustrate a DC motor control case, considering the field voltage as one of the control inputs in the domain of a continuous function U(*t*).

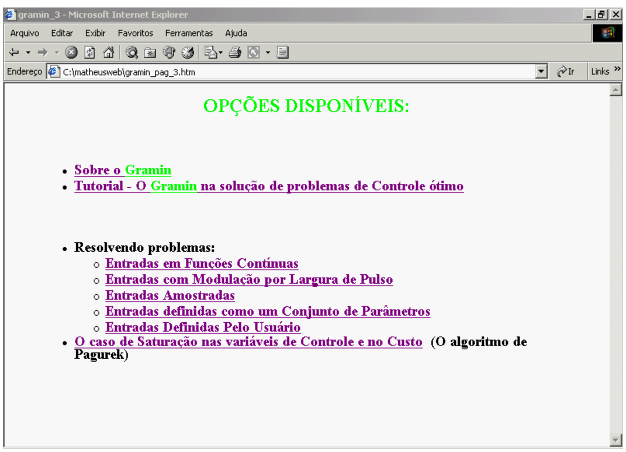

Fig. 1: GRAMIN menu with the available options

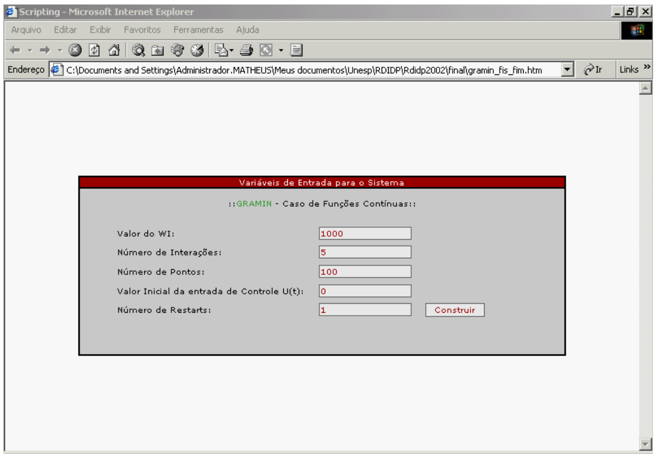

Fig. 2: User definition of the initial parameters

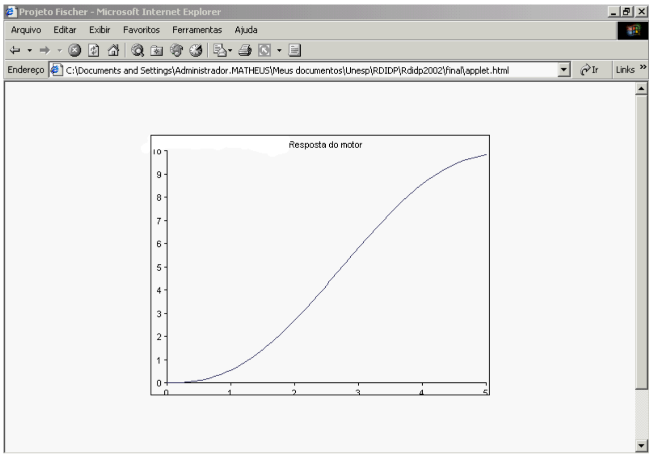

Fig. 3: User window – position of the motor axis

4. Conclusion

The remote implementation of the GRAMIN application is an useful resource for supporting the teaching of optimal control solving via the CGM, as it hides from the user all the programmning details associated with numerical methods. Besides this advantage, the implement solution protects the source code as it resides in the server side. This alternative is suitable for certain applications where the provider charges a certain amount from the users (i.e. pay-per-use), who would not have access to a copy of the application as it would be running at the server side.

5. References

[1] Hestnes, M.R.; Stiefel, E. "Conjugated Gradients Methods for Solving linear Systems". J. Res. N.B.S. vol 49, p. 409, (1952).

[2] Fischer, B. R. "Análise e Implementação de Algoritmos de Controle Ótimo pelo Método do Gradiente Conjugado". Dissertação de Mestrado: Faculdade de Engenharia Elétrica, Unicamp, Campinas-SP, Brasil, 1983, 170p. http://libdigi.unicamp.br/document/?code=vtls000048126