An Applet for Optimal Control Problems

RUI GONÇALVES, MARIA DO ROSÁRIO DE PINHO, JOÃO BORGES SOUSA
Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotecnica e de Computadores
Rua Dr. Roberto Frias s/n, 4200-465 Porto
PORTUGAL
http://www.fe.up.pt

Abstract: An Optimal Control Applet (OCA) is presented. This OCA is an Applet designed to study optimal control problems in an interactive manner. The graphical user interface (GUI) facilities the interaction with the user in several ways. First, the user is able to specify the key ingredients of an optimal control problem, namely controlled differential equations, control constraints and cost functions. Second, the user is allowed to specify specify the control inputs. These interactions are designed to support the cognitive process of learning the fundamentals of optimal control. The application is available for students over the net.

Key–Words: Computers, Internet, Multimedia in Engineering Education, New Technologies in Education.

1 Introduction

Teaching and learning optimal control is a tough endeavor. Optimal control theory is generally considered an advanced topic. The reason for this may come from the fact that any thorough course on optimal control assumes an advanced background not making it an appealing subject to newcomers on the field. In particular, undergraduates students when first faced with optimal control problems find it hard to build some sensitivity and insight to such problems, even when they turn to the available software tools for help. This may be because these lack some fundamental geometric insights. The geometric interpretation behind some of the main concepts in optimal control may come as a surprise, and also as a discovery. This interpretation seems to have been somewhat forgotten as the tools were getting more complex and more technical.

To reduce the "entry" cost into the field of optimal control, we should perhaps explore some of the initial developments in the field when the concepts were simpler, but powerful enough to result in the extraordinary development of this field since the sixties.

One way to accomplish this is through experimentation with the basic entities associated to an optimal control problem. Such experimentation was not easy until the advent of computers with powerful computational and graphical capabilities, and the availability of software packages capable of handling the intricacies of numerical and symbolic solvers for differential equations and the visualization of the corresponding trajectories. Since we believe it is important to develop new tools to assist students building a good insight of optimal control problems we developed OCA, an Applet for learning optimal control. The OCA is complemented with a Wiki for a shared learning experience. It is being used in tutored sessions for novice users.

The OCA we report on in this paper permits the user to experiment with the basic entities associated to an optimal control problem: controlled differential equations, control constraints and cost functions. Moreover, it allows the user to specify control functions, parameterized over time, for the system. Under proper guidance, the user becomes familiar with essential non-intuitive aspects of optimal control techniques:

- ○ non-linear behavior of the trajectories with respect to variations in the control inputs;
- ○ behavior of the cost for limiting behavior of pairs controls-trajectories;
- ○ variation of the qualitative behavior of the solution due to changes of initial conditions.

These aspects of experimentation lead the interested user to an enquiry into the reasons behind these nontrivial behaviors. It is time then to go into the technicalities behind optimal control.

There are many simple implementations in Java$^{\text{TM}}$Applet[5] format over the Internet used to guide students through math problems and techniques[9]. Some optimal control books are accompanied by CD-ROMs with applications and examples[6]. Also, some books already support the modern teaching aids available, having an companion

Website [3] with Java$^{\text{TM}}$Applets. To the best of our knowledge, the OCA has some innovations that make it a user friendly application. Notably it allows the user to explore different controls policies for a wide variety of problems.

The paper is organized as follows. In section 2 we describe the optimal control problem formulation. In section 3 we present the Applet from the user's point of view. In section 4 we describe the architecture of the application. In section 5 we describe a case study of the tool and in section 6 we draw the conclusions.

2 The Optimal Control Problem

OCA can handle optimal control problems of the following form:

$$\begin{cases} \text{Minimize } G(N(b)) + \displaystyle\int_a^b L(t,N(t),U(t))\,dt \\ \text{subject to} \\ \begin{array}{rcl} \dot{N}(t) & = & F(t,N(t),U(t)) \\ U(t) & \in & \Omega(t) \\ N(a) & = & N_0 \end{array} \end{cases} \tag{1}$$

with data the set $[a,b]$, functions

$G : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$,

$L : [a,b] \times \mathbb{R}^n \times \mathbb{R}^k$,

$F : [a,b] \times \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$,

a multifunction $\Omega : [a,b] \to \mathbb{R}^k$ and a given vector $N_0 \in \mathbb{R}^n$. Here n, the number of state variable $N(t)$, and k, the number of control variables $U(t)$ are both assumed to be greater or equal to 1.

3 Main Features of the Applet

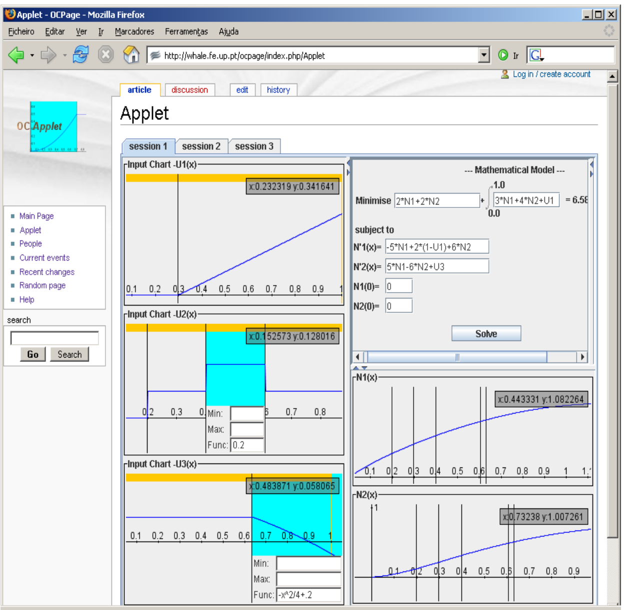
The main technical requirements for OCA are listed next:

- Interface with a symbolic expression solver;
- Handle ordinary differential equations with several variables.
- Small footprint to be rapidly served over the Internet.
- Automated PDF reports with the description of the given inputs, dynamic, model and final cost.

All this requirements are contemplated in a Java$^{\text{TM}}$Applet context. Symbolic processing of differential equations and integrals is done with the recourse of Matlab. A Web Server (see [7] for details) was created to serve as bridge between OCA and Matlab. A graphical Java2D component is used to support the specification of the control inputs by the user (see [4]). The control inputs are defined in a piecewise fashion over intervals defined by the user. A symbolic expression evaluator library [8] is used by this chart component to make the function representation.

The combination of these technologies made it possible for OCA to have a small 220KB footprint and to serve it over the Internet in any browser (Fig. 1).

OCA allows the user to define each optimal control problem (1), that is, the interval $[a,b]$, the cost, the dynamics and the initial state N_{n_0} (or $N_n(a)$). It also permits the definition of the multifunction $\Omega_k(t)$, assumed to be piecewise constant. The Applet is able to process models with several input functions $U_k(t)$ with $k \;>=\; 1$. Also the number of state variables $N_n(t)$ can be chosen with $n >= 1$. Once the optimal control problem is defined, the user can test different control policies. The control has to be a piecewise continuous function. For each control policy chosen the Applet plots the graph of each component of the state variable $N_n(t)$ and calculates the corresponding cost. Different control policies can be tested for the same problem in different sessions and the results can be compared.



http://whale.fe.up.pt/ocpage/index.php/Applet

Applet

--- Mathematical Model ---

Minimise 2*N1+2*N2 + ∫ 0.0 1.0 3*N1+4*N2+U1 = 6.58

subject to

N'1(x)= -5*N1+2*(1-U1)+6*N2

N'2(x)= 5*N1-6*N2+U3

N1(0)= 0

N2(0)= 0

Min:
Max:
Func: 0.2

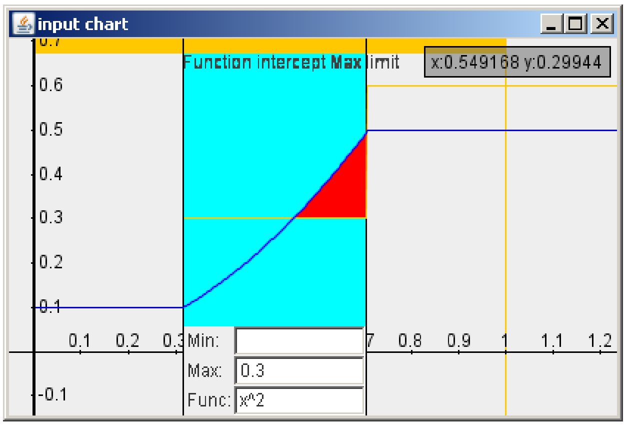
Min:
Max:
Func: -x^2/4+.2

Fig. 1: OCA running inside a Media Wiki (in this case:k=3 inputs and n=2 state variables)

4 Applet Architecture

The Applet user interface is divided in four areas: inputs charts, model definition, result dynamics charts and debugging output. The inputs for optimal control problem are made using a Java2D chart component, developed specially for OCA (Fig. 2). For each input function of the model, a chart panel component is used. In this chart panel the user/student defines a subinterval of the function domain by simply clicking with the right mouse button over the chart component. This creates a function delimiter where the user specifies the function inside the text boxes (symbolic e.g. "tˆ 2", "sin(t)/2", "log(t)", etc) and the graph of the written function is automatically displayed on the chart.

Constraints can be added to this input function by simply defining a maximum and minimum limit function [1]. The component notifies the user about constraints violations. Also, the chart component is used to make the representation of the state variable evolution (dynamics) of the model. In this case, the input events are turned off. Internally the Applet fills the charts panels with the processed functions. The processing domain of the OCA [a,b] is also defined in this panel, represented by the yellow top bar. The user can change the limits of this interval by dragging the extremities of the yellow bar with the mouse. Processing the user's model with the given inputs involves handling a system of differential equations. From the start the OCA intends to work in a symbolic philosophy. We chose to use Matlab for symbolic processing. The connection of OCA to Matlab is made through a Web Server. On the Web Server (Server side) there is a Java^{TM}application connected to Matlab running in singleton. This application executes the remote Matlab commands supplying the server which return the responses to the Applet in SOAP protocol (Fig. 3). Each time the user presses the 'solve' button in the model definition area several remote calls are made to the Web Server, depending on the model and inputs complexities. OCA has been tested with models of varying complexity and serving multiple users. The response time has been acceptable and the server remains stable.



Function intercept Max limit

x:0.549168 y:0.29944

Min:

Max: 0.3

Func: x^2

Fig. 2: Input chart component

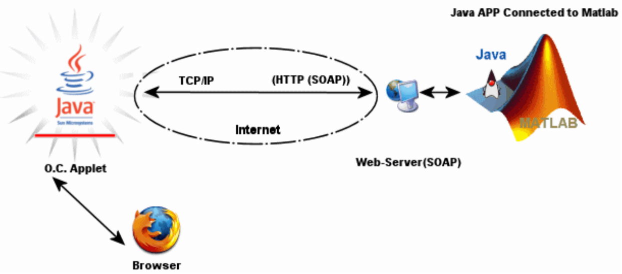


Fig. 3: Connection to Matlab architecture

To take interaction to a more academic level the Applet was integrated inside a Wiki page. The Optimal Control Wiki is based on MediaWiki^{TM}and it was parameterized to accept pure HTML on its pages. That way we include the OCA inside one of the Wiki pages (Fig. 1). As normal, major part of the Optimal Control Wiki is permitted to be editble by the Users/Students.

5 An Example

Here, we describe the use of OCA with an example of a cancer chemotherapy model proposed in [1]. The problem of interest is:

$$\left\{\begin{array}{l}
\text{Minimize } r_1 N_1(b) + r_2 N_2(b) + \\
\qquad \displaystyle\int_0^b (q_1 N_1(t) + q_2 N_2(t) + u(t))\,dt \\
\text{subject to} \\
\begin{array}{rcl}
\dot{N}_1(t) & = & -a_1 N_1(t) + 2a_2(1 - su(t))N_2(t) \\
\dot{N}_2(t) & = & a_1 N_1(t) - a_2 N_2(t) \\
u(t) & \in & [0,1] \\
N_1(0) & = & N_{10} \\
N_2(0) & = & N_{20}
\end{array}
\end{array}\right.$$

In this model the cell cycle are divided into two compartments; 1 and 2. The states N_1 and N_2 denote the number of cancer cells in each compartment. The scalar control $u : [0,T] \longrightarrow [0,1]$ represents the drug dosage administered with $u(t) = 0$ corresponding to no treatment and $u(t) = 1$ corresponding to a maximum dose. The objective is to minimize the number of cancer cells at the end of a fixed therapy interval without killing the patient. Since the drug can produce harmful effects on the patient and it is advisable to keep the number of cancer cells low during the

[1] This is why the multifunction $\Omega(t)$ is assumed to be piecewise constant.

whole treatment. The cost has two terms:

$$r_1 N1(b) + r_2 N2(b)$$

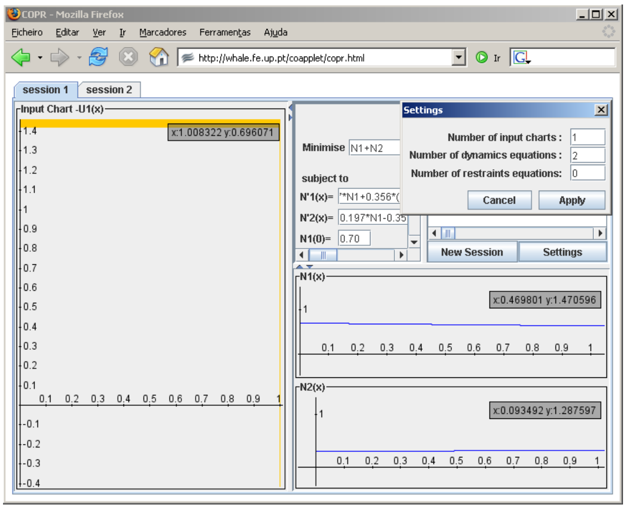
represents a weighted average of the total number of cancer cells at the end of the therapy and the integral

$$\int_0^b (q_1 N_1(t) + q_2 N_2(t) + u(t))\, dt$$

models the cumulative negative effects of the treatment and the weighted average of the total number of cancer cells at each instant. The parameter s is an effectiveness factor of the drug and it is assumed to be $s \geq 1/2$. The parameters r_1, r_2, q_1, q_2 are positive. Also, a_1 and a_2 are constants related to the type of cancer and the individual. We refer the reader to [1] for a complete description of the model.

5.1 Model Type

This problem has one input function ($k = 1$), two state variables ($n = 2$) (no state constraints). The user sets these parameters by first pressing the button "settings" on the upper right panel of the Applet. This action opens a dialog box where the user edits the main settings of problem; number of input or control charts (in this case $= 1$), number of dynamics equations ($= 2$) and no state constrains. See Fig. 4.


COPR - Mozilla Firefox

http://whale.fe.up.pt/coapplet/copr.html

Minimise N1+N2

subject to

N'1(x)= '*N1+0.356*(

N'2(x)= 0.197*N1-0.35

N1(0)= 0.70

Settings

Number of input charts : 1

Number of dynamics equations : 2

Number of restraints equations: 0

Fig. 4: Setup of the OCA main parameters

5.2 Definition of the Model

As in [1] we choose $b = T = 1$, $a_1 = 0.197$, $a_2 = 0.356$ and $s = 0.8$. As initial conditions we take $N_{10} = 0.70$ and $N_{20} = 0.30$. In the cost model we consider $r = (1, 1)$ and $q = (1, 1)$. These are the cancer chemotherapy parameters used in the mathematical model panel of the OCA. E.g. the "N1'(t)=" (i.e. $\dot{N}_1(t)$) textbox is filled with the string:

"-0.197*N1+2*0.356*(1-0.8*U1)*N2"

Here "N1", "N2" and "U1" correspond respectively to $N_1(t)$, $N_2(t)$ and $U_1(t)$. Since the OCA works in a symbolic way there is no need to define the "integration step" parameter.

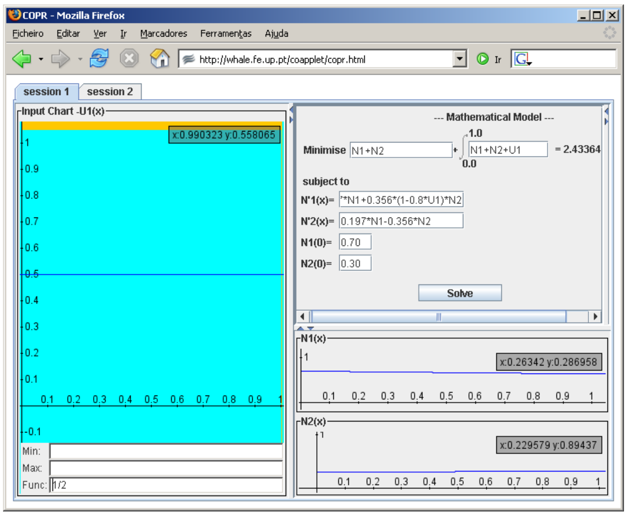
5.3 Inputs

For this problem we test two different control policies. First, we take

$$u_1(t) = \frac{1}{2} \text{ for } t \in [0, 1]. \tag{2}$$

As shown in [1] this is a *singular* control.

The definition of the controls in OCA is done in the input chart (described in section 4). The user has to create the function delimiters (in this example (2) it is only one), drag them to define the right interval and write the function on the "Func:" text box ("1/2" in this case). A pointwise control constraint $u_1(t) \in [0, 1]$ (i.e., here $\Omega_1(t) = [0, 1]$) can be defined as well. The user has to fill the "Max:" and "Min:" fields. After the model and input definitions is complete the user presses the "solve" button. The results, i.e., the cost and the graphical representation of the state trajectories appear on the OCA as shown in Fig. 5.


COPR - Mozilla Firefox

http://whale.fe.up.pt/coapplet/copr.html

--- Mathematical Model ---

Minimise N1+N2 + $\int_{0.0}^{1.0}$ N1+N2+U1 = 2.43364

subject to

N'1(x)= '*N1+0.356*(1-0.8*U1)*N2

N'2(x)= 0.197*N1-0.356*N2

N1(0)= 0.70

N2(0)= 0.30

Min:

Max:

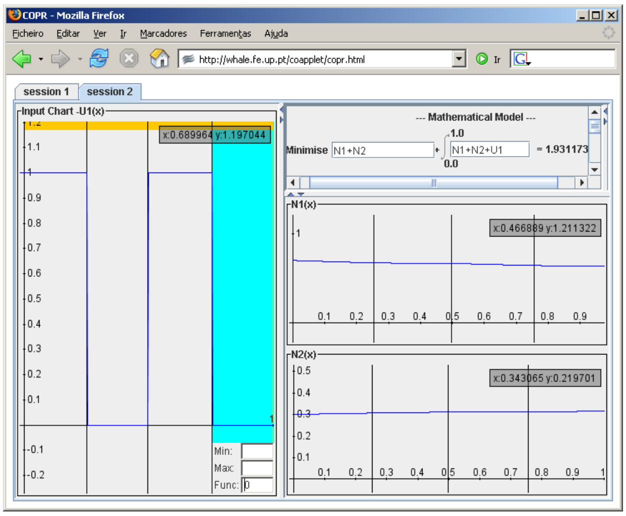
Func: 1/2

Fig. 5: Cancer chemotherapy model with a singular control ($U_1(t) = 1/2$)

Next, we test the following bang-bang control

$$u_1(t) = \begin{cases} 1 & \text{if} \quad t \in [0,\, 0.25[, \\ 0 & \text{if} \quad t \in [0.25,\, 0.5[, \\ 1 & \text{if} \quad t \in [0.5,\, 0.75[, \\ 0 & \text{if} \quad t \in [0.75,\, 1]. \end{cases} \tag{3}$$

We want to keep the results of the previous experimentation. To do so we need to click the ”New Session” button. A new separator will be created with a copy of the previous data session. Now the user only needs to introduce the new input policy (3). This is done by changing the input function $U_1(t)$ in the input chart; new delimiters have to be placed and defined. The results are shown now in Fig. (6).



Minimise N1+N2 + $\int_{0.0}^{1.0}$ N1+N2+U1 = 1.931173

Fig. 6: Cancer chemotherapy model with a bang-bang control

By clicking on ”Session 1” separator the user has access to the data of the previous session and the results can be compared. In fact, comparing the results from both sessions we deduce that although the patient receives the same amount of drug over the interval $[0, 1]$, the bang-bang control produces a lower cost (1.93) than the singular control (2.43). In view of [1] this comes as no surprise.

6 Conclusion

The OCA we report on this paper is a valuable tool to help the students understanding optimal control problems. It can handle both linear and nonlinear problems with various variables. The OCA permit the user to test different control policies for a chosen optimal control problem. It is worth mentioning that the OCA can be easily modify to estimate reachable sets for a given control problem.

The main setback of this Applet is that it cannot solve the optimal control problem. We aim to improve this Applet by introducing a solver routine. Once this is done, the OCA can be a powerful tool to determine if a certain admissible solution is sub-optimal.

We also hope to improve this Applet so that it can optimal control problems with state constraints and mixed state-control constraints.

Acknowledgements: The research was supported by FEDER and Fundação para a Ciência e Tecnologia, FCT, project POSC/EEA-SRI/61831/2004.

The authors thank Maria Margarida Ferreira, Fernando Fontes and Fernando Lobo Pereira for many discussions on the subject.

References:

[1] M.d.R. de Pinho, M.M.A. Ferreira, U. Ledzewicz and H. Schaettler, A model for cancer chemotherapy with state-space constraints, *Nonlinear Analysis* 63, 2005, e2591-e2602.

[2] V. Girault and P.A. Raviart, *Finite Element Methods for Navier–Stokes Equations,* Springer –Verlag, Berlin–Heidelberg–New York–Tokyo 1986

[3] G.C. Goodwin, S. F. Graebe and M.E. Salgado, *Control System Design* , Prentice Hall, 2000.

[4] V. J. Hardy, *Java 2D API Graphics* , Prentice Hall, 1999.

[5] K.C. Hopson and Stephen E. Ingram , *Developing Professional Java Applets* , Sams, 1996.

[6] G. Smirnov and V. Bushenkov, *Curso de Optimização*, Escolar Editora, 1998.

[7] C. Taylor and T. Kimmett, *Core Java Web Server* , Prentice Hall, 1998.

[8] JEP - Java Math Expression Parser, < *http://www.singularsys.com/jep*>, 2006.

[9] Online supplement to Math–University of British Columbia, <*http://www.ugrad.math.ubc.ca/coursedoc/math101*>, 2007.