Web-Based Interactive Simulation of Control Experiments

Julian Kolodko, Abraham Kaithayil George, Stjepan Blazevic, Nanbin Wang, and Ljubo Vlacic
Intelligent Control Systems Laboratory (ICSL)

Griffith University

Nathan Q 4111, Australia

j.kolodko@griffith.edu.au, kaithayil.abraham@student.gu.edu.au,
stjepan.blazevic@student.gu.edu.au, nanbin.wang@student.gu.edu.au,
l.vlacic@griffith.edu.au

Abstract – This paper deals with web assisted control education. It gives a taxonomy of the resources available on the internet and provides examples for each class. The paper then gives a more concrete example of online control education by describing a recently developed interactive simulation of autonomous vehicle manoeuvres and a range of other control problems designed to highlight particular aspects of control theory.

1. INTRODUCTION

The internet has become a powerful medium for communication and interaction between people and devices around the globe. It provides us with a medium to control processes and devices connected to the internet across the world. This paper is about the utilisation of these properties for control education. First, a review and categorisation of existing online control education facilities is given, then we discuss our contributions to control education: a web based platform for simulation of manoeuvres of autonomous vehicles and a web assisted control education site. This system has been developed as part of a larger web based control education concept proposed by the Intelligent Control Systems Laboratory (ICSL).

This paper is organised as follows. First the structure of a typical online control facility is presented in section 2. Section 3 reviews currently available online control education websites. In sections 4 and 5 we consider our control education sites. A discussion of the issues is given in section 6.

2. TYPICAL ARCHITECTURE OF AN ONLINE LAB

Figure 1 shows the typical architecture [1] of an online control facility. The principal components of such a facility are: (i) the Internet which provides a communication medium; (ii) a web browser which provides a user interface; and (iii) Java applets which provide control of a process. Packages like Matlab or Mathematica and a database do the background processing [2]. The final component is the experimental device. In the case of simulated experiments, Java applets are used to provide animations and simulations.

The online lab follows a client - server architecture. Students login from with their PC to the server through Internet. The server provides the concurrent access for multiple users if the

Fig. 1. Typical architecture of an online Lab

experimental apparatus allows. An audio-video server could be incorporated to provide the real time experience and a virtual presence. A web browser provides the user interface. Java applets are loaded from the server when they are required.

Where real time operation is required, the server can use a real time operating system.

3. REVIEW OF CONTROL EDUCATION WEBSITES

3.1 Requirements of Internet based systems -

An excellent review of the internet based systems is presented in a recently published paper by Dormido [3]. The primary recommendations from that and other papers (e.g. [4]) can be summarised as follows:

- An online lab should be accessible to the user any time, he or she decides to use the online facility.
- The user should be able to access the resources on the online facility with out having to purchase or install any special software.
- The online facility should be easy to navigate and use, in the absence of a supervisor.
- The Internet based system should hold the interest of the student and should provide the feeling of actually being present at the experiment facility.
- The content of the site should use up to date technology and course material.

Before presenting our taxonomy of online control education resources, we refer the interested reader to the EDCOM (IFAC Technical Committee on Control Education) web site at <http://www.gu.edu.au/centre/icsl/edcom>. In the "project initiatives" section of the website you will find links to example sites for each category as follows:

3.2 Control Education Websites

These educational websites are based on distance education concept. Students can gain knowledge in control theory at their convenience, on the Internet. Interactive course material allows students to hold their interest. The content of these educational websites can be purely theoretical or it may relate to some experimental setup or perhaps a combination of theory and practise.

3.3 Virtual libraries, databases, references, etc.

Such websites hold vast information as e-journals, training materials, databases, references, etc and provide links to professional societies, university and college departments, information services, research projects and recruitment agencies.

3.4 Online Labs

Providing laboratory facility to distance education students of control engineering had been a problem in the past. They had to be present in the lab to perform the experiments. Now online control laboratories provide students access via the Internet to various experiments in control engineering. They can login from anywhere and perform experiments at their convenience. Students can observe dynamic phenomena that are difficult to comprehend. Unique or expensive equipment can be shared between different Universities and a wider range of experiments can be made accessible to the students. Online experiments could be simulated or with real hardware.

Simulations - The need to integrate traditional classroom lecture with some kind of experimental practise is often addressed using simulation tools. This is a cost effective and safe way to allow students to interact with the systems. But some physical realities are omitted. It can never substitute for experiments with actual systems.

Experiments with real hardware - These provide students access to live real time experiments over the Internet. Experiments with real hardware are costlier and face various constraints. But, this is the best and closest way to substitute a real life laboratory experiment.

Tele-operation of devices connected to the Internet - Tele-operation involves interaction between a human operator and remote device via Internet. The remote device could be anything ranging from webcams to robots. The control of robots remotely over the Internet is called Tele-robotics.

The EDCOM website also provides information on web assisted learning concepts developed through joint work between universities. Such cooperation greatly reduces infrastructure cost since each participating University provides one or two experiments online but total number of experiments becomes much greater than a single university could afford otherwise.

4. AUTONOMOUS VEHICLE MANOUVRES

The intelligent Control Systems Laboratory has developed a number of mobile robot based test-beds for use in evaluating cooperative autonomous vehicle concepts without the need for large scale testing facilities. These robots features (i) a distributed multi-microcontroller based architecture; and (ii) multi-sensor sub-systems that include infrared and ultrasonic ranging sensors, optical flow sensors, radio packet communication system and a laser based sensory system. A number of driving manoeuvres have been developed for these test beds including intelligent speed adaptation, stop & go motion control, lane keeping, intersection navigation, overtaking, distance control, tele-operation of multiple platforms, etc. The solutions for these driving manoeuvres have been developed on the grounds of both Decision & Control Theories and a Sensor Fusion Paradigm.

Currently we are moving towards establishing an internet based tele-operation of these platforms to make them accessible to researchers world wide. In addition we have designed a web based platform for interactive simulation of the autonomous vehicle manoeuvres as a bridge from interface design to true tele-operation. We now discuss that interactive simulation environment and cover two manoeuvres that have been developed.

The simulation environment has been developed with Java applets and html files as explained in figure 1. The implementation requires no additional software or hardware on the user side. The standard browsers support java applets and are down loaded on to the user's computer as the webpage is opened. The simulation consists of two parts. In the first part, distance control and overtaking was implemented. In the second part, unsignalised road intersection traversal crossing was implemented.

4.1 Distance control and overtaking manoeuvres

This part was implemented with Java applets and html file. Fig. 2 shows the web page implementation of the simulation. It has two tasks: (a) to keep the distance between two vehicles or mobile robots while cruising and (b) overtake of one vehicle by the other. Thus the platform can be used for easy demonstration of object tracking control and steady state error control. This is a practical visual experiment that allows students to better understand underlying control principles.

Fig 2: Webpage implementation of the simulation

Part a. The distance is set by the user and the two vehicles cruise along keeping that distance. The user enters a meaningful distance, say 50 – 100m, in the text box provided on the web page and clicks on the distance button. The distance between the two vehicles is set to that distance. When the user clicks on the start button, the vehicles start moving in the track, keeping the set distance.

The end user can also examine a quality of the set distance control by fine-tuning the controllers' parameters. After the user defines the control algorithm he or she can press a "submit" button, which causes the control parameters to be sent to a Matlab web server, which then passes the values onto Matlab software for calculation of the resulting system output. Since this data is also stored in a database, the end user can review a record of all past output in order to analyse quality (e.g. step response value) of the controller.

Part b. When the user clicks overtake button, the vehicle that is following the other in same lane at a fixed distance, moves to the next lane and starts to over take the first one. When the overtaking vehicle gets ahead of the first vehicle and reaches the fixed distance, it switches back to the original lane. Then the vehicles move along keeping the same fixed distance between them; until the user again clicks on the overtake button.

4.2 Unsignalised road intersection traversal

This simulation demonstrates how intelligent vehicles should behave at an unsignalised road intersection or junction and cross it without a collision. It demonstrates the use of the Constraint Satisfaction Theory and the Decision Making

Fig 3: Webpage implementation of crossing manoeuvres

Theory. The webpage implementation of the simulation is shown in fig. 3.

The intelligent vehicles have priorities assigned to them. When they come to a junction or intersection, the vehicle with the higher priority crosses and the other vehicles wait for the crossing vehicle to clear the junction. Thus they wait for their turn to cross. In the simulation, a higher priority is set for a vehicle either manually, by clicking on one the buttons provided or automatically, ie, on the basis of 'first come first go', which means that a priority is conferred to the incoming vehicle, relative to its distance from the intersection.

4.3 Implementation Details

4.3.1 Distance control and overtaking manoeuvres

When the web browser accesses the html page containing one of our simulations, it loads the appropriate java applet from the server which is initialised using the Int() method. In this method, the variables for vehicle position, speed, and direction are initialised. The setDistance() method is called by java script from the html file when the distance button is clicked. The distance in the text field of the html file is passed on as argument to the setDistance method.

When the run button on the web page is clicked, the java script calls the startAnimation() method of the applet. It in turn repaints the applet with new values of vehicle position and direction. The vehicle positions are calculated in the calculateNextPos() method.

When the OverTake button is clicked on the webpage, the java script invokes the overTakenow() method. This method keeps track of which vehicle is overtaking. It invokes another method, findDiff() to find the difference in vehicle positions. The run() method translates the vehicle positions and directions and repaints the paint() method of the applet. The hasOverTaken() method keeps track of the overtaking and flips the vehicle positions when they achieve the set distance.

4.3.2 Crossing manoeuvres

The priority of crossing is set by clicking on the corresponding buttons. A vehicle is given highest priority if the user clicks on the button (Pink or Cyan) corresponding to that vehicle. The user can also set the priority to First In, First Out by clicking the FIFO button. These buttons pass 0, 1 and 2 values, corresponding to the button, to the setPriority() method of the applet. Here again, the java script interfaces between the applet and the buttons.

When the start button on the webpage is clicked, the java script calls the simulate() method of the applet. The resolvePriority() method keeps track of which vehicle has the priority of taking the crossing. The calculateNextPos() method calculates the movement of the vehicles. The hasCarCrossed() method ensures the lower priority vehicle waits while the high priority vehicle crosses the intersection.

The paint() method is the simulation engine of the applet. For each position of the vehicles the method repaints thus giving the impression of continuous movement of vehicles.

5. OTHER ANIMATED EXAMPLES

5.1 Site Overview

Our experience shows that real life animated examples from each control systems teaching area enhances the students' understanding of control systems methodologies and adds to the students ability to visualise particular system parameters and the effect of their changes on the systems' output response. It was found that a web-assisted learning concept, as developed by Griffith University's Intelligent Control Systems lab (www.griffith.edu.au/centre/icsl/vlagic/teaching) provides undergraduate students with a new, refreshing, online, very fast and successful way of learning of the basic control systems techniques, such as: Time response, Steady-State error, Stability, Root-Locus, Nyquist and Design (PI, PD, PID).

To develop its animated web-assisted learning site, the ICSL utilised Macromedia Flash technology to create animations and Macromedia Dreamweaver and Microsoft Frontpage to create a cross-browser compatible web site. A detailed explanation of the animation design process (using a particular example) can be found in [5][6][7].

The site is structured around the broad control system techniques mentioned above (see figure 4) with each section explained via a tutorial example. These examples consist of a problem definition, solution and problem animation and are formulated as per the text from which they were sourced ([8][9][10]).

Figure 4. Web site structure

Figure 5. Animation Examples

Tutorial examples were chosen to provide a full and clear presentation of the particular Control Systems technique, of the problem requirements and of the obtained solution results. Also, a very important requirement was that the tutorial examples were related to real world engineering problems (elevator, helicopter, an industry robot, a liquid storage tank, etc., see figure 5). The selected tutorial examples were aimed at providing the students with an easy, fast, effective and enjoyable way of learning and fully understanding of the particular Control Systems technique. As the Control Systems website gives the end-users an opportunity for a real-time multi-user access all examples' Matlab code design solutions were written in a such way that the user can simply change particular control system parameters (K , $OS\%$, ω_n , etc) from the keyboard and observe the changed parameters' effect on the design solution. All Matlab codes were also written in a logical order (program flow) with a comment and an explanation of the used Matlab function in each line, giving the user an opportunity to understand the whole design process with a minimal effort.

Each example was animated with an aim to reinforce, clarify and provide full understanding of the particular Control Systems methodology.

6. DISCUSSION

Regards the simulation of autonomous vehicle manoeuvres, there remain two problems to be resolved: (i) when the vehicles are around the corner, and the user clicks overtake, the vehicles go out of control; this is due to the fact that calculations in vector space have not yet been implemented; (ii) the applet doesn't respond properly to clicking on distance button while the vehicles are moving. The distance has to be set before the vehicles start moving. These will be resolved in the near future.

We also find that a visual demonstration of the simulation experiments needs to be enriched towards demonstrating what may happen if the underpinning theoretical concept were not implemented.

7. REFERENCES

- [1] Rohrig, Christof and Andreas Jochheim (2002). "Java based framework for remote access to laboratory Experiments" *Advances in control education 2000*. Edited by L. Vlacic and M. Brisk. Pages: 67-72 Publisher: Pergamon.
- [2] Kovacs, Ferenc, Kristztian Monostori, Hassan Charaf, Ruth Bars and Robert Tuschak (2000). "Utilising Internet in teaching Control theory". *Advances in control education 2000*. Edited by L. Vlacic and M. Brisk. Pages: 55-59 Publisher: Pergamon.
- [3] Dormido, B. S. (2002). "Control learning: Present and Future" *Plenary papers, Survey papers and Milestones*. Pages: 81-103. IFAC 15th world congress. Barcelona 2002.
- [4] Chen Y., F. Naghdy and J. Rogers (2000). "A web based intelligent tutoring system for Instruction in a control system laboratory". *Advances in control education 2000*. Edited by L. Vlacic and M. Brisk. Pages: 61-66 Publisher: Pergamon.
- [5] Blake, Bonnie (2001): *How to Do Everything with Macromedia Flash 5*. Osborne / McGraw-Hill.
- [6] Leete, Gurdy & Ellen Finkelstein (2002): *Macromedia Flash MX for dummies*. Hungry Minds.
- [7] Ulrich, Katherine (1999): *Visual Quickstart Guide Flash 4 for Windows and Macintosh*. Addison-Wesley.
- [8] Dorf, Richard C. & Robert H. Bishop (2000): *Modern Control Systems*. Prentice Hall.
- [9] Kuo, Benjamin (1995): *Automatic Control Systems*. 7th ed. Prentice Hall International.
- [10] Nise, Norman S (2000): *Control Systems Engineering*. 3rd ed. John Wiley & Sons.