

Learn2Control: A Web-based Framework for Project-Oriented Control Education

Thomas Tometzki* Marten Völker** Christian Blichmann*
Ernesto Elias-Nieland*** Christian Sonntag*
Sebastian Engell*

* *Process Dynamics and Operations Group,
Dept. of Biochemical and Chemical Engineering,
Technische Universität Dortmund, 44221 Dortmund, Germany
{t.tometzki | c.blichmann | c.sonntag | s.engell}@bci.tu-dortmund.de*
** *Bayer Technology Services GmbH, 51368 Leverkusen, Germany
martin.voelker@bayertechnology.com*
*** *West Pacific Consulting Group, Burnaby, BC V5J 3M6, Canada
ernesto.elias-nieland@wpcg.ca*

Abstract: Important skills of a control engineer are the ability to subdivide complex control design projects into smaller design steps and to solve the subproblems, taking into account the interdependencies between the subtasks as well as the overall goals and requirements. In this paper, the web-based learning environment *Learn2Control* is presented which has been developed to complement classical teaching methods in the control engineering education at Technische Universität Dortmund. *Learn2Control* provides the students with the opportunity to apply their knowledge of control theory and to gain experience in project-oriented workflows by means of authentic case studies on control systems design. The didactic concept aims at teaching the dependencies and the interactions between the solution of the subtasks of modeling, analysis, and controller design, and between the methods used for these tasks. In the latest version of *Learn2Control*, only a standard web browser and Java have to be installed on the client computers. The modeling, analysis and design tasks are processed on web pages which are generated by Java server technology on a web server. For mathematical operations, a custom multi-user *Matlab* web service was developed. Currently, three control design projects are available within the new framework. In addition, four projects are available in a previous version of the framework that requires a *Matlab* installation on the client side.

Keywords: e-learning, control education, project-oriented learning, software architectures

1. INTRODUCTION

The traditional control education is characterized by the presentation of a coherent theoretical background and of a large variety of mathematical techniques for the solution of subproblems of an overall controller design process such as system identification, controller design, and simulation. The students are able to apply predefined methods to well-defined subtasks of practical problems, but usually there is little opportunity to tackle the solution of larger, not pre-structured problems. The necessary skills for the solution of these complex problems include the partition of the overall problem into smaller tasks, the choice of suitable methods to address these subtasks, the choice of the right sequence of steps, the choice of a suitable level of effort that is invested in the solution of the subproblems, and the consideration of the dependencies of the subtasks and the overall goals and constraints of the project. These skills are not provided by the classical control education. The laboratory experiments that are carried out in the control education often involve the solution of a sequence of predefined and well structured tasks and can usually not be performed in an autonomous fashion.

Nowadays, supplementary computer-based learning via the internet is provided by several academic groups. Using computer-based tools for calculations and simulations, visualizations of virtual plants, and remote control of real laboratory experiments, control engineering can be conveyed to the students more efficiently. However, in our view the potential of internet-based learning is not yet fully used. Modern technology offers the potential to provide computer-based experiments where complex practical control engineering problems that consist of several interdependent and not pre-structured subtasks have to be solved.

The e-learning environment *Learn2Control* aims at providing the opportunity to acquire knowledge and experience with project-oriented work by means of authentic control system design projects. The first version of *Learn2Control* [Learn2Control, 2008] (initially called *PROLERN*) was developed in a collaborative effort by research groups from the Institute of Automation and Computer Control at Ruhr-Universität Bochum, the Process Dynamics and Operations Group at Technische Universität Dortmund, and several other German research groups [Schmid et al., 2001]. In subsequent years, it was improved and extended

considerably (see Völker et al. [2004]). While the first version of *Learn2Control* offers a large variety of possibilities to the students, it suffers from the drawback that it is essentially client-based, i.e. it requires client installations of several (commercial) tools, such as e.g. *Matlab*. This paper presents a redesign of *Learn2Control* that was recently developed by the authors and that is completely server-based. Seven projects have by now been realized in *Learn2Control*, most of which are focused on control systems design. Two of these projects include remote experiments in which the users have to collect process measurements and can test the designed controllers on real plants via the internet.

2. THE DIDACTIC CONCEPT

The emphasis of the web-based learning environment *Learn2Control* is on project-oriented learning of methods for practical control engineering. The implementation of the projects is based upon a learning model which describes all possible solution paths as well as the prerequisites for performing the individual steps by a hierarchical task graph.

2.1 Project-oriented control education

Most universities are using a classical approach to teach control techniques and concepts to future engineers in which the students go through a number of courses that are supplemented by a small number of laboratory experiments. This approach equips the students with a sound foundation to deal with dynamic systems and control problems, but does not provide the skills required for the solution of larger modeling and control tasks. In our project-oriented approach, learning goes beyond the classroom practice of short and isolated tasks and instead emphasizes learning activities that are long-term, interdisciplinary, and integrated with real-world issues and practices. One immediate benefit of project-oriented learning is that it can significantly increase the motivation of the students by engaging them in their own learning. The students have to apply and to integrate different pieces of knowledge within the solution of a challenging problem. The concept of project-oriented learning is the distinguishing feature of the *Learn2Control* platform.

All projects within *Learn2Control* start from a detailed description of the problem at hand and a specification of the overall goals of controller design and of the limitations that have to be taken into account during the design process. Based on this information, the students have to find a suitable strategy to solve the task. They have to decide which subtasks, e.g. modeling, controller design, simulation, etc., have to be performed, and in which sequence the subtasks are tackled. On the subtask level, different methods are offered, and the students have to decide which method they apply. This may lead to a dead end and to the necessity of returning to a previously reached state in case of unsuitable sequences. The return to a previous state in the design process can also be necessary to improve the design, to compare different parameter settings used in a step, and to compare different methods to solve the subproblems.

2.2 The hierarchical task graph

In order to guide the problem solving process, a learning model was developed which describes the subtasks within a project, including all interdependencies between them, on an abstract level. In *Learn2Control*, the learning model is formulated as a hierarchical task graph which consists of nodes that represent the subtasks as well as directed edges that represent interdependencies. The nodes can be nested. As an example, a node may contain different subnodes that represent subtasks, some or all of which have to be finished to complete the primary task. Each node is associated with a web page that handles and guides the processing of the subtask. Directed edges represent the interdependencies between two nodes on the same hierarchical level. These directed edges indicate possible paths through the task graph. A directed edge between two nodes defines that the successful completion of the predecessor node is a precondition to work on a successor node. Additionally, feasible paths are indicated by colors: disabled nodes are shown in red color, accessible nodes in blue, and finished nodes/tasks in green. The work on a project can be represented as a path through the task graph, starting from a predefined node in which the design problem is specified, and ending at a node in which e.g. the control performance is validated by experiments or simulations. Besides an optimal solution path, less favorable or even infeasible paths may exist. In contrast to classical learning environments in which the user often has to proceed in a predefined linear fashion, the *Learn2Control* approach supports the free combination of subtasks and the modeling of the interdependencies between these subtasks.

For modeling and testing of task graphs, an authoring tool has been developed using the meta-modeling environment DOME [DOME, 2008]. Two code generators have been integrated into DOME which automatically generate a Java applet of the task graph, the so-called *task machine*. Within DOME, the modification and the creation of nodes and directed edges of the *task machine* can be performed via drag-and-drop techniques. During a design project, the *task machine* supports and controls the actions of the users on the project web pages, and a suitable representation of the *task machine* provides comfortable navigation within the project web pages and represents the current project status. Figure 1 shows a DOME screenshot that represents a part of the task graph of the Heating System Project (see Sec. 4.1). The node "Modeling" contains three different subnodes. The subtask "Modeling" can be finished only if "Theoretical Modeling" or "Experimental Modeling" and subsequently "Validation" have been solved.

3. THE NEW LEARN2CONTROL SOFTWARE ARCHITECTURE

After some experience with the previous client-server integrated architecture [Völker et al., 2004], it was decided to redesign it as a fully server-based architecture employing cutting-edge web technologies. Most of the code had to be rewritten and some of the existing projects were retrofitted to the new environment. In contrast to the previous version, the need for local installations of complex (licensed) programs on the client side is not needed any

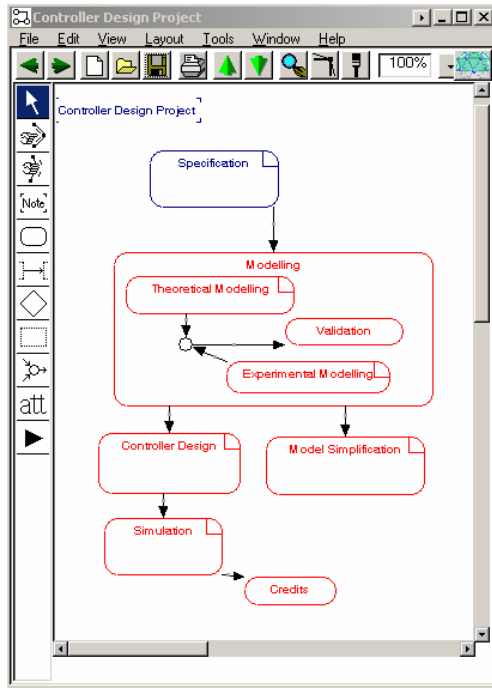


Fig. 1. A partial task graph for the heating system project.

more. The use of *Learn2Control* requires not more than a standard browser and Java, and no additional local installations are necessary. An overview of the new server-based architecture is shown in Figure 2, details are given in the following subsections. The connections of the server to the web pages are indicated by arrows on the left side. The arrows on the right side indicate the three components that must be implemented using authoring tools during the design or the extension of a project.

3.1 The application server

A basic module of the *Learn2Control* framework is the application server (Apache Tomcat) that delivers applications to the client computers. The server runs servlets that execute e.g. client authentication, session handling, and the *task machine* object that models the projects and their status. Each running servlet is an autonomous object that receives a request (e.g. from a client computer) and generates a response based on that request.

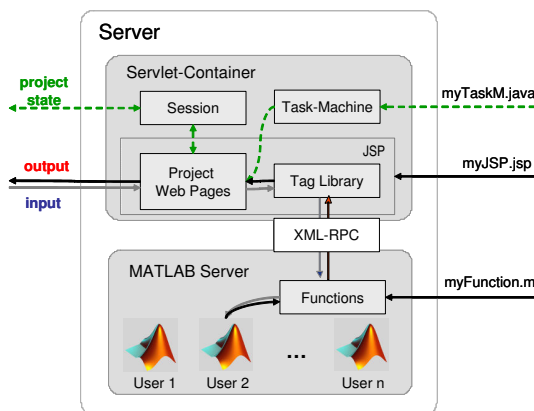


Fig. 2. Structure of the new server-based architecture.

3.2 Java Server Pages (JSP)

The main benefit of an application server with servlets (denoted as servlet container) is the ease of development of applications: servlets can be programmed using Java Server Pages (JSP) which offer the full flexibility of the Java language. This technology allows to generate web pages or other types of documents dynamically in response to a web client request. The distinct feature of JSP is the possibility to hide the Java functionality behind Java Custom Tags that can be used like HTML tags. Hence, the authoring process does not necessarily require Java programming skills. A first Java Custom Tags Library (see Tag Library in Figure 2) has been created which offers basic functionalities. From the markup point of view, writing JSP code is similar to writing HTML code. From the business logic point of view, however, JSP is very different from HTML and Java in that the complete logic resides on the server. In our projects, we use a model-view-controller (MVC) approach to implement the logic within the projects. In the MVC, three different JSP files are used for one subtask: the model JSP describes the processing functionality, the view JSP generates the web page for the client, and the controller JSP combines processing and visualization.

As mentioned, JSP are used that enable the use of custom tag markups. So far, all necessary *Matlab* interfacing operations are encapsulated in custom tags that are included in the Tag Library. Furthermore, custom tags for entering and displaying transfer functions are available. Therefore, the HTML markup of a web page can be written using any text-editor; specialized HTML editors (e. g. WYSIWYG editors) can offer convenience and additional functionality.

3.3 The task machine

For each user who is logged on to a particular project, a *task machine* object is created as a servlet on the server side. A local copy of the Java program, an applet in this case, is downloaded to the client if either the project status or the currently active project web page have been changed (see "project state" arrow in Figure 2). To visualize the project state and the current subtask for the user, the applet has visualization capabilities that have been realized in Java2D.

3.4 Multi-user administration

The new version of *Learn2Control* realizes a true multi-user environment. Projects can use custom classes for session handling and user administration. Currently, there are two kinds of users: administrators and normal users. Administrators have access to the configuration backend, a special administration site written and implemented in JSP, and can add, delete, and modify users and project assignments. Normal users have the ability to log on to the *Learn2Control* environment and to use the projects they are assigned to. User data and configuration data is stored internally in a MySQL [MySQL, 2008] database.

3.5 The Matlab workspace server

For computationally demanding operations, a custom *Matlab* workspace server is used that provides the func-

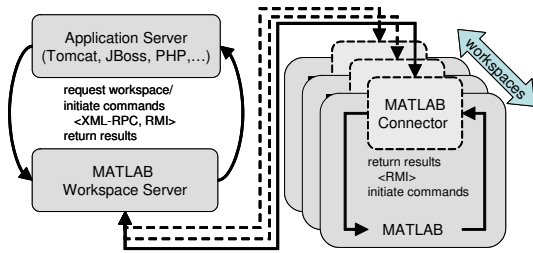


Fig. 3. Overview of the Matlab workspace server.

tion of managing multiple *Matlab* workspaces on a single physical server. Each user session has its own workspace that is completely isolated from the other workspaces. In the workspaces, calculations can be performed simultaneously, in contrast to the default single-threaded behavior of the *Matlab* web server [Mathworks, Inc., 2002] that supports only one thread at a time. Furthermore, the workspaces serve as storage containers for temporary results. An overview of the architecture of the *Matlab* workspace server is given in Figure 3. The session JSP (or any other software that supports either the RMI or the XML-RPC protocol) requests a new workspace session via the programming interface, and can execute arbitrary *Matlab* commands and return results and error messages. Internally, the workspace server manages a pool of *Matlab* workspaces which are created on demand. Each active workspace corresponds to a running *Matlab* process.

As on modern operating systems, the system memory for the required linked libraries is usually shared among multiple processes, this is an efficient approach to provide a multi-threaded, multi-workspace environment. On a moderately equipped server (Athlon64 3800+ dual-core with 3.5GB of RAM), up to 20 workspaces were already operated in parallel successfully.

3.6 Remote experimentation

The *Learn2Control* software architecture includes the option to integrate remote experiments into the projects. A database system schedules the time slots of the remote experiments and controls the access of the users. The students have to book experimentation time. The user interface that is necessary to control the experiments is organized and provided by the experiment server. Figure 4 shows an example user interface taken from the project for twin rotor control system design which is developed in cooperation with the Fachhochschule Köln. The twin

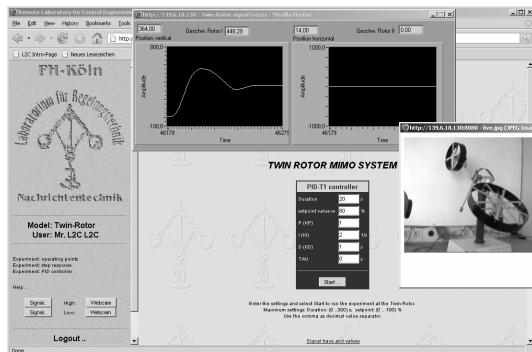


Fig. 4. A screenshot of the twin rotor experiment.

rotor project includes a remote real-world experiment that can be used to validate identification models and to test controllers at the real plant.

4. LEARN2CONTROL PROJECTS

This chapter briefly describes three *Learn2Control* projects that have already been implemented in the new framework. These projects are routinely used in our mandatory course "Process Control Laboratory" (4 CP) that consists of nine practical lab experiments. It takes place in the second semester of the Master Programme "Automation and Robotics" at Technische Universität Dortmund. Before each experiment, the students are required to pass a short examination on the underlying principles. During the experiments and particularly in a closing discussion, the results are reviewed by the students and the supervisors. In addition, four older projects are provided. These are described in [Völker et al., 2004].

4.1 Controller design for the heating system of a reactive distillation column

This project deals with the design of a controller for the heating system of a reactive distillation column in which a heterogeneously catalyzed esterification of methanol and acetic acid for the production of methyl acetate is performed. In the bottom of the column, an evaporator produces a steam flow that enters the column. The steam flow rate is adjusted by the controller of the heating system. It is an important manipulated variable of the process.

In the heating system, thermal energy is transferred to the bottom of column by the circulation of electrically heated water. The flowchart of the heating system is shown in Figure 5. In order to solve the task to design a suitable controller for the heating system, the students must complete several subtasks for which several possibilities are provided. The first step is to understand the system and the requirements for the performance of the controller which are specified by the typical indicators: rise time, transient time, and integral square control deviation. Thereafter, the students can decide between two different paths through the modeling part, experimental or theoretical modeling. The corresponding task graph for this part of the project is displayed in Figure 1.

In the experimental modeling section, the students must first determine a suitable control structure, i.e. choose the measured and manipulated variables of the system. The

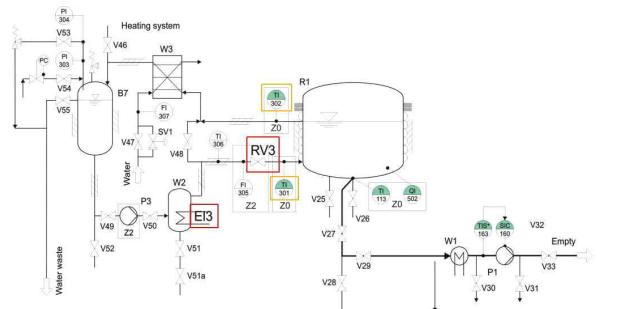


Fig. 5. Flowchart of the heating system.

next step is to define a test signal that excites the process in a way that it is possible to identify the dynamic behavior of the process. It is recommended to first determine the dominating time constants of the system. To this end, the students have to inspect step responses that are obtained from two experiments in which steps are applied to one of the two inputs of the system while the other input remains zero. Based on the experimental results, the students must select the signal types, the sampling times, the durations, and amplitudes of test signals for system identification. The students have the opportunity to define a large variety of signals which can be saved. In the next step, the students must perform a simulation of the rigorous reference model using the identification signals. The output data of the simulation is automatically saved and can be loaded in the following identification step. The students can choose between various identification schemes: ARMA (Auto-Regressive Moving-Average), ARX (Auto-Regressive with eXogeneous input), Box-Jenkins, and Output-Error models can be identified. After completion of the identification task the resulting model can be saved and used in following project tasks.

The theoretical modeling procedure realized in this project consists of three steps which can only be performed sequentially. In the first step, the correct homogeneous compartments of the system have to be chosen via a dropdown menu. In the following step, the state variables and the external values of the heating system must be defined. Finally, the differential and algebraic equations that describe the system must be determined and entered. After the complete input into a HTML-box, the system of equations is transferred to *Matlab* and checked for syntax errors. If it is syntactically correct, the implicit differential-algebraic-system is transformed into a semi-explicit differential-algebraic system. The node "Validation" enables the user to validate the model against the reference model.

After the students have solved the modeling task, they can choose between the simplification of an existing model or the design of a model-based controller. In the latter case, the students can select several design methods, e.g. the frequency domain design method that uses frequency responses and Bode diagrams. After entering the corresponding web page, the students must select and load an existing model from the dropdown list that has been generated in a previous modeling task. The frequency domain design method requires a linear model, and if a non-linear model is loaded, an error message occurs and gives the advice to either load a linear model or proceed with the simplification in order to derive a linear model. After the successful import of a linear model, the structure and the parameterization of the controller are shown, and several input fields are displayed into which the user has to enter his choice of parameters (Figure 6). After the submission of the values Bode plots show the open-loop frequency responses (Figure 7). The students can modify the parameters of the controller and observe the resulting Bode plots. The students can also save several controllers within the Matlab data model. After the design of one or more controllers, the processing is continued with the validation of the controllers. The students can check if the designed controllers comply with the overall requirements using simulations.

Fig. 6. Input page for the controller parameterization.

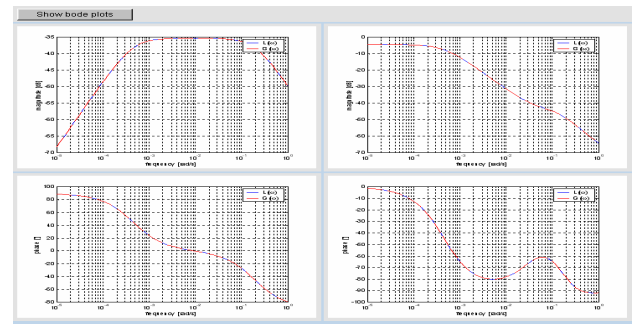


Fig. 7. Bode diagram in the controller design step.

4.2 FASTER - A tool for the synthesis of multivariate low-order controllers

FASTER is a tool for synthesizing multivariate low-order controllers by means of frequency response approximation. The original idea is described in [Engell, 1988], and the extension to the multivariate case has been published first in [Engell and Müller, 1991, 1992]. In a recent paper, the application of FASTER to a complex controller design case study, the control of a pilot-plant scale reactive distillation column is described [Völker et al., 2007]. The algorithm minimizes the difference (in terms of the Frobenius norm of the frequency responses) between the desired closed-loop behavior and the behavior of the closed-loop with a reduced-order controller of a predefined structure, leading to a nonconvex optimization problem. For certain standard controller structures, optimal controller parameters can be computed by solving a convex problem. By fixing some controller parameters, good starting points for the nonconvex optimization problem can be computed.

In the FASTER project, the design of a controller by frequency response approximation is modeled as a three-step procedure. Figure 8 shows a web page associated with a subtask of the project. Each web page includes the graphical representation of the *task machine* at the top, an overview of the data structure on the left, the main section in the center, and a help section on the right. The FASTER project is defined by the *task machine* graph at the top in Figure 8. In the step "Enter Plant", a linear plant model must be specified. In the second step "Specify T0", the desired closed-loop behavior must be entered. To simplify the procedure, default models for the closed-loop behavior such as first-order and second-order transfer functions with delay are offered. In the

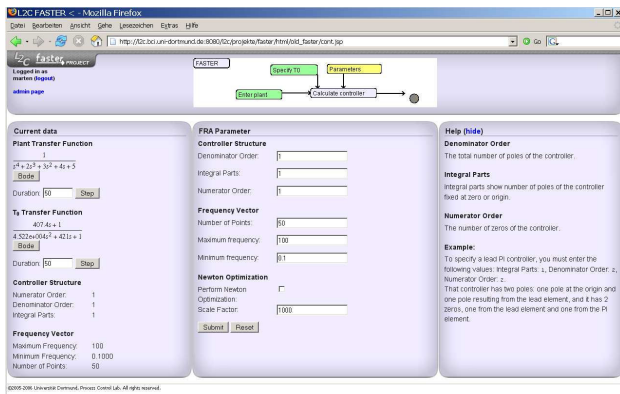


Fig. 8. A web page of FASTER.

step "Parameters", the controller structure, the frequency range for the approximation, and some other parameters must be chosen. Finally, in the node "Calculate controller" the controller can be calculated after all data has been collected.

On the left-hand side, an overview of the actual data is shown (this is done for all nodes of this project). This view provides the user with an overview of the plant, the desired closed-loop behavior, and the parameters of the approximation algorithm in a single figure. Additionally, step responses and frequency responses of the plant and of the desired closed-loop system can be displayed. On the right-hand side, a help window explains the meaning of the parameters for new users. If the help window is disengaged, the extra space is divided between the data overview section and the main section.

4.3 A fundamental course on system analysis

The goal of this *Learn2Control* project is to provide an opportunity to apply theoretical knowledge and to gain experience in basic principles for the analysis of dynamic systems. An illustrative example of a simple continuous stirred tank reactor system is considered in which an equilibrium reaction $A \rightleftharpoons B$ of two species takes place. The design of this project web page is similar to the design of the FASTER project and uses the same stylesheet. Since this project has the character of a tutorial exercise, the task graph has a linear structure.

The first task is to describe the dynamic system by means of three differential equations for the state variables, the concentrations of both substances and the volume of the content of the reactor. The next step is to calculate the steady state of the system as a prerequisite of a subsequent linearization around this point. By consideration of the eigenvalues of the resulting Jacobian matrix, a statement about the stability of the system can be made. Thereafter the transfer function of the system must be computed from the previous linearization. For the resulting transfer function, the root locus diagram can be displayed. Step responses can also be plotted.

5. CONCLUSIONS AND FUTURE WORK

The emphasis of the learning environment *Learn2Control* is on web-based project-oriented training of the application of methods that can be applied to solve practical

control engineering tasks and of the solution of larger design problems that involves several different steps. The projects are realized using a task graph that encodes all possible solution paths as well as the prerequisites for their completion in a hierarchical task graph. The data generated during the execution of a project is stored and saved in a data structure that enables the users to store more than one solution. The differences in the performance that result from changes of the parameters or from the use of different design methods can be observed. Possibly, certain design steps have to be repeated in order to meet the specifications. In contrast to traditional learning environments, *Learn2Control* helps to understand the dependencies and the interactions between the various methods and tasks for control system design and therefore enables the users to gain experience in structuring control engineering projects.

To increase the usability of *Learn2Control*, a completely new software architecture was designed. The modifications concern a server-side integration of Matlab, the *task machine*, the programming of the project web pages in JSP, and the multi-user administration with a database connection. *Learn2Control* is used in our engineering education within the Master Programme "Automation and Robotics" at Technische Universität Dortmund. *Learn2Control* is continuously updated and improved.

REFERENCES

DOME. *The DOME Modeling Environment*. Honeywell International Inc., Morristown, NJ, USA, 2008. <http://www.htc.honeywell.com/dome>.

S. Engell. Compensator design by frequency-weighted approximation. *Proc. IEE International Conference on Control*, pages 253–258, 1988.

S. Engell and R. Müller. Controller design by frequency-weighted approximation: The multivariable case. In *Proc. IEE International Conference CONTROL-91*, pages 581–586, 1991.

S. Engell and Ralf Müller. Fast and efficient selection of control structures. *Computers and Chemical Engineering*, 16:157–164, 1992.

Learn2Control. *Process Dynamics and Operations Group, Technische Universität Dortmund, Germany*, 2008. <http://l2c.bci.uni-dortmund.de:8080/l2c>.

Mathworks, Inc. *MATLAB External Interfaces*, Natick, MA, USA, 2002.

MySQL. *MySQL GmbH, München, Germany*, 2008. <http://www.mysql.com>.

C. Schmid, R. Bartz, N. Becker, S. Engell, H. Roth, and H. M. Schaedel. PROLERN - A project-oriented approach to teaching of control engineering. In *Proc. IFAC Workshop on Internet Based Control Education (IBCE'01)*, 2001.

M. Völker, A. Liefeldt, S. Engell, and C. Schmid. Learn2Control: A project-oriented approach to teaching control engineering. In *Proc. IEEE Conference on Computer-aided Control System Design (CACSD)*, pages 184–189, 2004.

M. Völker, C. Sonntag, and S. Engell. Control of integrated processes: a case study on reactive distillation in a medium-scale pilot plant. *Control Engineering Practice (Special Issue on Award Winning Applications)*, 15(7): 863–881, 2007.