Learning Dynamics and Control Using Remotely Tutored Simulation and Virtual Experiments

H. Mann, M. Sevcenko
Czech Technical University / Computing and Information Centre, Prague, Czech Republic

Abstract—**The *Dyn*LAB project developed by an international consortium aims at motivating young people to engineering study, and at improving engineering training using innovative didactic and technological approaches. The resulting web-based course is supported across the Internet by a software environment including a robust DYNAST simulation engine, publishing and monitoring tools, and a large collection of re-solvable examples including 3D virtual experiments. DYNAST solves nonlinear algebro-differential equations submitted in a textual form. For a system model submitted in a graphical form characterizing the system real configuration DYNAST formulates the underlying equations automatically. It is also capable of providing the system semisymbolic analysis in time- and frequency-domains. Besides this, DYNAST can be also used across the Internet as a modeling toolbox for the MATLAB control-design toolset.**

Index Terms—**control, dynamics, e-learning, Internet, remote simulation, virtual experiments.**

I. Introduction

The subject of dynamics and control underlies all aspects of modern technology and plays the determining role in the world-market competition of engineering products. Its importance increases with the ever-growing demands on operational speed, efficiency, safety, reliability, or environmental protection of the products. Nevertheless, national authorities and entrepreneurs in many countries report lack of qualified engineers as well as a critical overall decline of interest in engineering study among young people. Professional associations call for radical changes in the engineering curriculum and for new innovative approaches to vocational training (e.g. [1]).

The existing courses on dynamics and control are criticized namely for

- discouraging young people from engineering study by overemphasis on theory and mathematics at the expense of practical engineering issues in the curriculum
- separating courses on dynamics analysis along the borders between the traditional engineering disciplines despite the fact that most of the contemporary engineering products are of multidisciplinary nature
- presenting ‘textbook’ problems carefully engineered to fit the standard ‘underlying’ theory without having the students to undertake realistic modeling
- using computers to carry out old exercises without radical modification of the curriculum to incorporate computers in a way fully exploiting their contemporary capabilities

II. Project DynLAB

A project called DynLAB – *Course on Dynamics of Multidisciplinary and Controlled Systems in a Virtual Lab* has been developed by an international consortium [2]. The emphasis and style of the course differs from most of the existing courses by a number of innovative features:

- exposing learners to a novel systematic and efficient methodology for realistic modeling of multidisciplinary system dynamics applicable to electrical, magnetic, thermal, fluid, acoustic and mechanical dynamic effects in a unified way
- introducing learners to the methodology through simple, yet practical, examples to stimulate their interest in engineering before exposing them to rigorous theory and advanced mathematics
- giving learners a better ‘feel’ for the topic by problem on-line simulation, graphical visualization, and by interactive virtual experiments
- allowing different target groups to select individual paths through the course tailor-made to their actual needs and respecting their background
- allowing both for self-study and remote tutoring with investigative and collaborative modes of learning
- integrating computers into the course curriculum consistently and giving learners a hands-on opportunity to acquire the necessary skills
- exploiting the computers not only for equation solving, but also for their formulation minimizing thus learners’ distraction from their study objectives
- giving learners the opportunity to benefit from ‘organisational learning’, i.e. from utilizing knowledge recorded during previous problem solving both in academia and industry

The intended target groups of the *Dyn*LAB course are students wishing to complement the traditional courses, distance-education students at different levels of study, practicing engineers as a part of their life-long learning as well as teachers intending to innovate the courses they teach.

III. Multidisciplinary dynamics

The engineering systems become more and more complex with regards both to the number of their components as well as to the variety of phenomena affecting their dynamics, either in a useful or undesirable way. The phenomena involved in system dynamics might come simultaneously from different energy domains treated traditionally by different engineering disciplines (electrical, electronic, magnetic, mechanical, fluid, acoustic, thermal, etc.). To be able to cope with the contemporary systems, engineering students should be introduced into an approach to system dynamics treating phenomena from different domains in a unified way. Such a unified approach into engineering study gives students a more comprehensive view of the real world.

There are two additional advantages to the unified courses on system dynamics besides the ability to cope with the contemporary systems. First, a properly planned curriculum which includes such a course avoids unnecessary duplication. This allows introducing more advanced material into the study program. Second, a unified course is consistent with the growing tendency of students to decide on their branch of engineering relatively late in their study. Such a course is also appropriate as a part of a program of continuing education for graduated engineers who received their degrees earlier.

To understand and predict the dynamic behavior of such systems as well as to design them, engineers resort to computer-assisted *modeling, simulation and analysis*. Introduction of these techniques allows engineers considering a larger variety of designed systems and for their thorough verification before the system prototypes are constructed and tested experimentally. Testing by simulation is the only option in cases where experimentation is too expensive or dangerous. Simulation also helps to maintain the systems, and in the case of a system failure it can be used to diagnose the failure cause. As computation techniques allow for introducing new products of higher-quality into the market faster, they help in acquiring higher profits.

Simulation is also a well proven *learning tool* helping to facilitate learners' comprehension of dynamics and control principles. A modern course on engineering dynamics should consider *multi-level modeling*. When designing a complex system, engineers resort to dynamic models of several levels of system *abstraction and idealization*. The design process usually starts by the conceptual design phase of the highest abstraction and aims towards the technological phase in which an assembly or some other way of system production is designed. Between these, also two intermediate design phases, both concerned about system dynamics, can be recognized.

In the *functional design phase*, interactions between the system components are assumed to take form of physically dimensionless signals, states and disturbances. Typically, the design of automatic control, of digital circuitry architecture, or of imbedded software is carried within this phase. The *physical design phase* is concerned about implementation of the system architecture, functions and signals in terms of physical phenomena and quantities. Both useful and undesired multi-domain physical effects are considered here in terms of energy transfer, accumulation and dissipation. The interrelationships of these effects are governed by physical laws.

IV. DYNAST software

A. Efficient simulation

In the past, *efficiency of simulation* was evaluated with regard to its demand of computer time only. Nowadays, however, the computer time is so inexpensive that the cost of simulation is dominated by the cost of personnel required to prepare the input data, to supervise the computation and to interpret the results. Therefore, an efficient simulation software tool should minimize demands on its users' time and qualification. In the other words, software should be sufficiently user-friendly and computationally robust.

When evaluating simulation software one should take into consideration its application area. Models of high abstraction and idealization used in the conceptual design of control, for example, can be conveniently represented by block diagrams. Using such block diagrams for physical-level models is, however, a cumbersome and error prone task. It requires an involved manual formulation of the underlying equations and, in addition, manual construction of block diagrams representing the equations.

B. DYNAST simulation software

To comply with the above mentioned efficiency requirements, the DYNAST software package was chosen as the kernel tool for multidisciplinary system simulation within the DynLAB course.

DYNAST provides there:

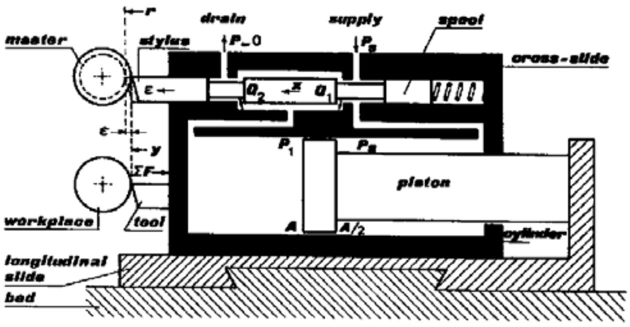
- solution of nonlinear differential and/or algebraic equations submitted in a natural textual form
- simulation of real dynamic systems the models of which are submitted in a graphical form resembling the real system configuration (the underlying equations are then formulated automatically)
- semisymbolic-form transfer functions and responses of automatically linearized system models
- online support for simulation, virtual experiments, and monitoring of submitted tasks
- modeling toolbox for MATLAB and Simulink

C. Multipole modeling

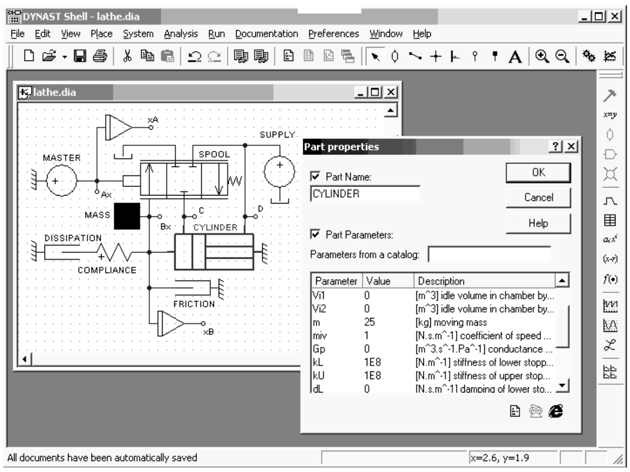
The automatic formulation of equations for physical-level modeling of multidisciplinary systems is based in DYNAST on *multipole modeling.* Such a modeling procedure starts with decomposition of the modeled systems into disjoint *subsystems*. This idea is similar to free body diagrams in mechanics or control surfaces in thermodynamics, for example. A subsystem multipole model approximates the subsystem energy interactions with the rest of the system under the assumptions that

- the interactions take place just in a limited number of *interaction sites* formed by adjacent *energy entries* into the subsystems (like fluid inlets, electrical terminals, translating or rotating mechanical connections, heat-transferring contact surfaces, etc.)
- the energy flow through each such entry can be expressed by a product of two complementary *power variables* (force – velocity, torque – angular velocity, volume flow rate – pressure, electrical current – voltage, magnetic flux rate – magnetic voltage, or entropy flow – temperature)

Each energy entry into a subsystem is represented in its multipole model by a *pole* associated with a pair of the power variables. In graphical symbols of individual multipoles, the poles are denoted by *pins*, i.e. by short line segments sticking out of the symbol outlines. Dynamics of a complete system is represented graphically by a *multipole diagram* consisting from symbols of subsystem multipole models. The sites of energy interaction between adjacent energy entries are portrayed in the diagram by the diagram *nodes*. The energy entries interacting mutually are represented in the diagram by pins interconnected to the same node by line segments called *links.* The links can be viewed as idealized subsystem interconnections capable of transferring energy in both directions without any dissipation, accumulation or delay.



(*a*)



(*b*)

Figure 1. *(a)* Copying lathe, (*b*) its multipole model.

In each energy domain, the most rudimental multipoles represent pure twopoles like pure energy sources, accumulators and dissipaters. Energy conversion from one domain to another one can be modeled by pure transducers. Only four different pure transducers are needed to model all kinds of electro-magnetic, electro-mechanical, magneto mechanical, fluid-mechanical, rotary-rectilinear, and other energy conversions. Multipole models of real subsystems like electronic or fluid devices, various mechanisms, motors or sensors, heating or cooling units, etc., can be build up from the pure multipoles. The multipoles can be combined also with blocks or equations.

Fig. 1 shows the cross-section of a copying lathe and the corresponding multipole model representing dynamics of the lathe mechanical and hydraulic components.

The most important advantage of multipole diagrams over block diagrams or bond graphs is in the *isomorphism* between their structure and the geometric configuration of the modeled real systems. In fact, multipole diagrams are mappings of real system representations from the geometric onto the topological space. The practical consequence of the isomorphism is that the multipole diagram can be set up in a kit-like fashion in the same way in which the real system is assembled from its subsystems. Such a modeling procedure can be based on mere inspection of the modeled real system. Recollect that a *block diagram* is just a graphical representation of a set of equations. The line segments interconnecting blocks are associated with just one variable that can propagate in one direction only.

D. Submodel libraries

DYNAST is accompanied by *libraries of submodels* for electronic and fluid-power devices, electro-mechanical transducers, mechanism parts, control units, etc. The submodel dynamics can be described by a combination of multipoles, blocks, and equations nested in a hierarchical way. The libraries are open for easy addition of user-defined submodels and their symbols. Each DYNAST submodel description is encapsulated in an independent file. The default values of submodel parameters can be overridden by values specified in terms of constants or symbolic expressions. Fig. 1 shows the dialog used to specify parameters of a hydraulic cylinder making a part of a machine.

Using the multipole approach is also of several other important advantages:

- multipole models can be developed, debugged, tuned up and validated once for ever for the individual subsystems independently of the rest of the system, and once they are formed they can be stored in submodel libraries to be used any time later
- this job can be done for different types of subsystems (e.g., fluid power devices, electronic elements, electrical machines, mechanisms, etc.) by specialists
- the submodel dynamics can be represented by different descriptions each of them suiting best to the related engineering discipline (lagrangian equations in mechanics, circuit diagrams in fluid power or electronics, block diagrams in control, etc.)
- modeling refinement or subsystem replacement (e.g., replacement of an electrical motor by a hydraulic one) can be taken into account by a submodel replacement without interfering with the rest of the system model

V. LEARNING ENVIRONMENT

A. Distributed simulation system

The DynLAB course is delivered within a web-based learning environment supporting learners' mutual collaboration and their communication with a tutor. Investigative learning is encouraged by a large collection of solved problems and virtual experiments. The examples can be resolved and modified in an interactive way across the Internet. This gives the learners a hands-on opportunity to acquire the necessary skills in solving real-life problems.

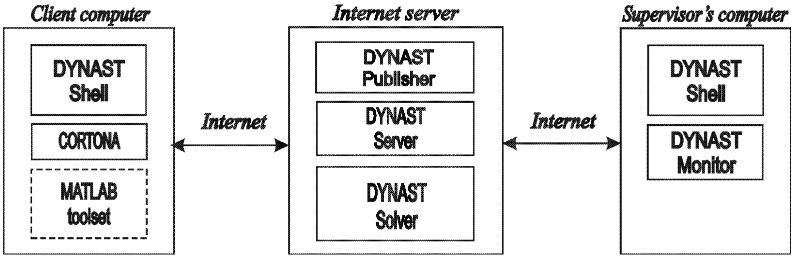


Figure 2: Environment for remote modeling, simulation and virtual experiments.

As shown in Fig. 2, the kernel of the *distributed simulation* system is formed by the DYNAST Solver that the learners can access across the Internet in several ways including even e-mail. Setting up multipole and block diagrams directly on a web page is enabled by the schematic editor DYNCAD formed by a Java applet. DYNCAD converts diagrams into the DYNAST input language and sends the data to the DYNAST Solver across the Internet. After the computation results are sent back and plotted on the client-computer screen.

Remote tutoring is enabled by the software tool called DYNAST Monitor that is linked to the DynLAB server across the Internet. It allows tutors to observe textual data and graphical-form diagrams submitted by learners to the DYNAST Solver. Tutors can not only monitor learners' activities, but they can also communicate with them, assist them in solving their problems and correct their errors if necessary. DYNAST Monitor appeared to be very useful even for tutors sharing the same computer room with learners.

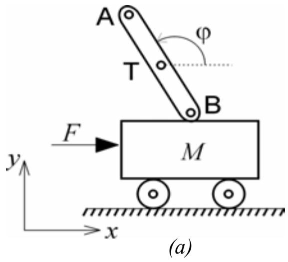
B. User-friendly simulation environment

Even more comfortable and user-friendly mode of access to DYNAST Solver provides DYNAST Shell. This mode requires, however, downloading and installing this free software on client computers with MS Windows. DYNAST Shell has been designed for a wide variety of tasks in a way suitable to users of different levels of qualification and experience. All operations are intuitive and they are supported by a context-sensitive help system.

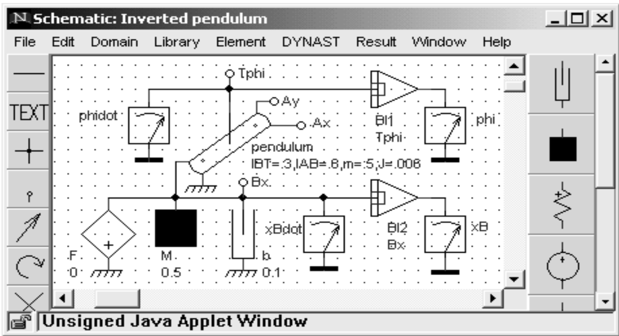
A built-in syntax analyzer is continuously checking the submitted data. Dialog windows (wizards) in DYNAST Shell allow for submitting data without knowledge of the input language (though DYNAST input language is very user-friendly and sounds natural to engineers). The input data is directly interpreted without any compilation delay. DYNAST Shell includes graphical editors for multipole and block diagrams as well as for submodel symbols. A special dialog box for each new submodel is formed automatically as shown in the screenshot of the DYNAST Shell graphical interface shown in Fig.1b.

DYNAST Shell can also communicate with the server-based DYNAST Publisher. It is a documentation system for automated publishing reports on simulation experiments and descriptions of library submodels using LaTeX. The systems extracts automatically the relevant parts of the input data and captures the submitted multipole or block diagrams as well as the resulting output plots and includes them into the documents. The documents can be converted by the server software into PostScript, PDF and HTML formats.

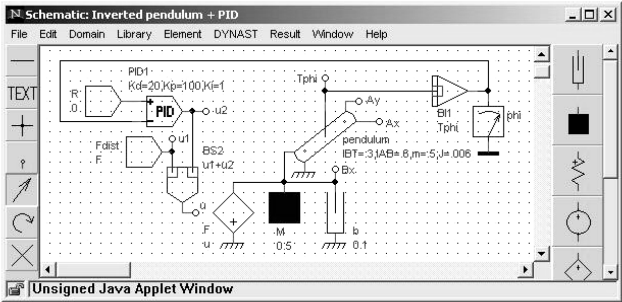
C. Support for control design



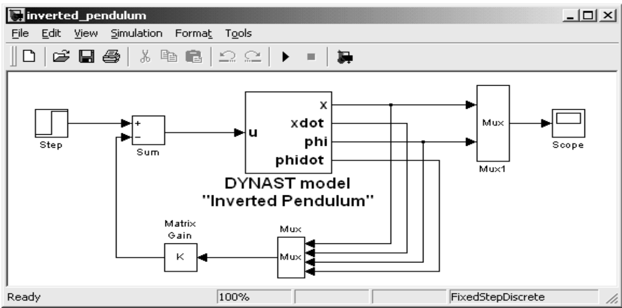
(a)



(b)



(c)



(d)

Figure 3: (a) Inverted pendulum, (b) multipole model, (c) analog PID control, (d) digital PID control.

In control design, the modeling efficiency of DYNAST can be combined to a great advantage with the control-design power of the MATLAB toolsets. Using DYNAST, a model of the plant to be controlled can be easily set up in a graphical form and then used to validate the open-loop model. At the same time, DYNAST is able to compute the required plant transfer-function poles and zeros and to export them to MATLAB in an M-file. When the control feedback loop is designed in the MATLAB environment and added to the plant model in DYNAST, the complete nonlinear control system is verified in DYNAST. In the case of digital control design, the control system configuration is implemented in Simulink. The block representing there the controlled plant remains, however, in DYNAST and communicates across the Internet with the rest of the diagram using the Simulink S-function.

Let us consider analog-PID control design for the plant in the form of the inverted pendulum given in Fig 3*a*. As it is shown in [4], a considerable number of manual operations is necessary before MATLAB can be exploited for computation of transfer functions for such a plant. DYNAST allows for avoiding all these tedious manual operations. Fig. 3*b* shows a multipole model of the pendulum, which has been set up using the web-based schematic editor DYNCAD. After verification of the plant open-loop responses using DYNCAD, an M-file with the plant transfer functions is exported across the Internet to MATLAB installed on a client computer. The plant PID control design by means of MATLAB can than proceed as described in [4]. Then the resulting feedback loop can be added to the plant model in DYNCAD as shown in Fig. 3*c*. Finally, the complete nonlinear control system is verified in DYNCAD.

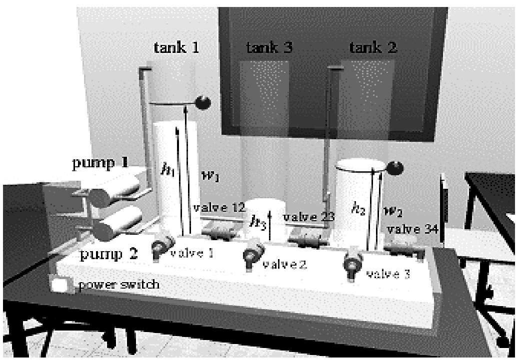
Also in the case of digital-PID control design the transfer-function data for the plant model is first exported to MATLAB. Then the digital control design can proceed as described in [4]. To verify the design in this case, the digital feedback loop is implemented in SIMULINK as shown in Fig. 3*d*. The large square block represents there the plant multipole model in DYNAST shown already in Fig. 3*b*. The communication across the Internet between this model remaining in DYNAST and the rest of the diagram implemented in SIMULINK is enabled by the SIMULINK S-function available for downloading at [2].

D. Virtual experiments

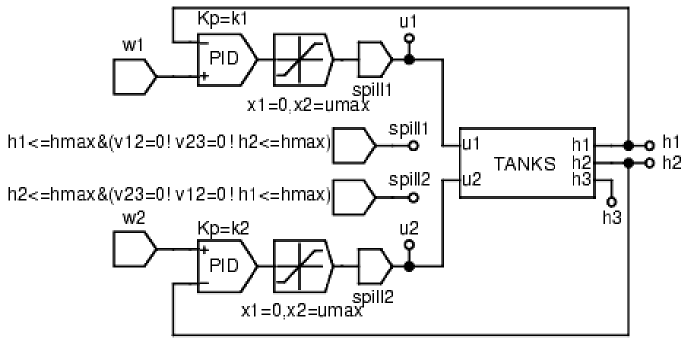
To stir up learners' interest in dynamics and control as well as to enhance their understanding of the topics, the course text is augmented by 3D virtual experiments, most of them interactively controllable. So far, the following experiments are available at the project website: Carriage & pendulum, Two tanks, Ball and beam, Gyro pendulum, Three tanks, VTOL (Vertical Take Off and Landing) aircraft emulator, Chemical reactor, Hydraulic cylinder, Optical tracker, Two-link planar robot. The motions of objects in all the experiments are driven by DYNAST across the Internet. The only software the learners need to download and install on their computers to be able to observe the experiments, is the Cortona freeware VRML browser.

As an example, Fig. 4a shows the three tank virtual experiment. By clicking the mouse over the screen of their computer students can adjust the red level marks on tanks 1 and 2, open or close any of the 6 valves interconnecting the tanks with each other or with an outlet, and they can switch on the pumps. This allows students to try to control the system manually in such a way that the levels in tanks 2 and 3 reach the level marks as soon as possible and stay there. Then they can go to the automatic control exploiting the default PID control, or they can test a control algorithm of their own design. The default control is illustrated by Fig. 4*b,* Fig. 4*c* shows the physical-level multipole model of the controlled plant. Submodels of motors, pumps, valves and open tanks are included in a lower hierarchical modeling level.

Fig. 5*a* shows a 3D virtual geometric model of a robot the motion of which is governed by DYNAST simulation across the Internet. The simulation utilizes the model of robot dynamics given in Fig. 5*b*. Besides the robot motion, learners can observe plotted responses of various robot variables like the arm trajectory shown in Fig. 5*c*, for example.

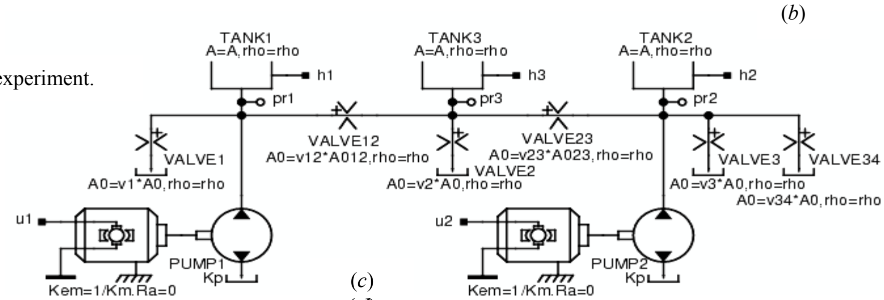


(*a*)

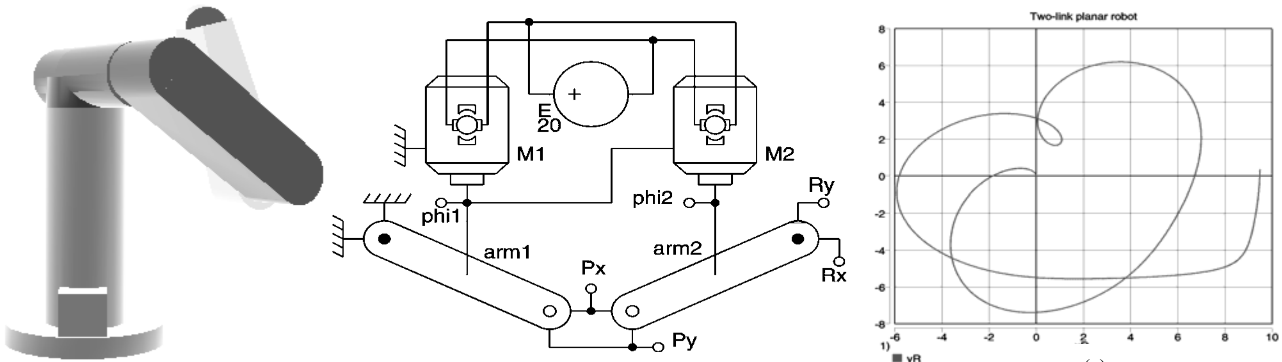


(*b*)

Figure 4. Three tank virtual experiment.



(*c*)



(*a*) (*b*) (*c*)

Figure 5: Robot: (*a*) virtual model, (*b*) dynamic model, (*c*) robot-arm trajectory.

VI. Learning modes

A short survey of different learning modes supported in *Dyn*LAB environment is given in Table 1. Novices are motivated to a more involved engineering investigation of system dynamic behavior by 'playing' with movable 3D virtual-reality models of various systems. They can change the model parameters and excitation while observing the model behavior not only qualitatively, but also quantitatively using virtual measuring instruments. In the next step, students can model, simulate and analyze dynamic behavior of given systems. The most advanced students are supposed to design controlled dynamic systems, to verify the designs and to optimize them.

TABLE I.
LEARNING MODES IN THE *DYN*LAB COURSE.

Learning objective	Prerequisites	Course assignment	
		Given	Task
stirring up interest in dynamics	high-school math and physics	3D virtual model of a real system	to modify system parameters or excitation and to observe changes in the system dynamic behavior
introduction to dynamic modeling	high-school math and physics	configuration of a real system	to set up the corresponding multipole diagram and to simulate the system dynamic behavior
more advanced dynamic modeling	fundamentals of system dynamics	configuration of real components	to set up multipole models and symbols of components, store them in a library, and validate their dynamic behavior
formulation of system equations	introduction to dynamic modeling	configuration of a real system	to form system equations, to solve them, and to compare the solution with the multipole-based simulation results
introduction to control design	formulation of system equations	plant specification & control objectives	to reduce the model, to design control, and to verify the design using the plant unreduced model
introduction to system design	introduction to control design	system specification	to design the plant as well as its control, then to verify and optimize the overall design
design of virtual experiments	advanced dynamic modeling	experiment specification	to design the virtual 3D geometric model, to set up the dynamic model, and to write the simulation script

Conclusions

The paper has presented a brief overview of a distributed software environment for a web based course on dynamics and control utilizing the Dynast modeling and simulation package. The course content as well as the software is available on the web at http://virtual.cvut.cz/dynlab/. Major features of the course are the viewpoint adopted the novelty of some of the examples, the ease of being able to study the effects of parameter variations and other aspects available with simulations created for user involvement; and the presence of some virtual reality experiments which can be operated manually or in closed loop.

Acknowledgements

The authors wish to acknowledge the support of the Leonardo da Vinci Programme in funding the project and the contributions to it of the other partners. In particular the contribution to the virtual reality experiments [5] by C. Schmid, the Ruhr-Universität partner, is greatly appreciated.

References

[1] "Future Directions in Control Education", *IEEE Control Systems*, Vol. 19, No. 5, Oct. 1999.

[2] Website of DynLAB, Pilot Project of the EU Leonardo da Vinci Vocational Training Programme, at http://virtual.cvut.cz/dynlab/

[3] Mann, H., M. Ševcenko: "Internet-Based Collaboration and Learning Environment for Efficient Simulation or Control Design". *Proc. Congress ASME, DSC* **71**, Washington D.C., 2003.

[4] Messner, B. and D. Tilbury: *Control Tutorials for MATLAB*, University of Michigan/ Prentice Hall 2000, http://www.engin.umich.edu/group/ctm/

[5] Schmid, C.: "A remote laboratory using virtual reality on the Web". *Simulation*, 73 (1999), 13-21.

Authors

H. Mann is with the Computing and Information Centre, Czech Technical University, Zikova 4, CZ-166 35 Prague 6, Czech Republic (e-mail: mann@ vc.cvut.cz).

M. Sevcenko is Prof. Mann's doctoral student (e-mail: sevcenko@ vc.cvut.cz).