

A WEB BASED COURSE IN CONTROL USING DYNAST

Derek P. Atherton¹, Heřman Mann²

¹*Dept. of Engineering and Design, University of Sussex, Falmer, Brighton,
BN1 9QT England, d.p.atherton@sussex.ac.uk*
²*Computing and Information Centre, Czech Technical University in Prague,
Zikova 4, CZ-166 35 Prague 6, Czech Republic, mann@vc.cvut.cz*

Abstract: The paper discusses the development of a first course in control engineering and its support using the simulation package Dynast. The work, which has several partners, was supported by the Leonardo Agency in a more extensive project DynLAB on web based education. Examples and virtual reality experiments in Dynast support the learning of the control analysis and design methods. *Copyright © 2005 IFAC*

Keywords: Education, simulation, virtual reality, controller, nonlinear control

1. INTRODUCTION

The paper describes the development of a web-based course on dynamics of multidisciplinary controlled systems DynLAB (see References for Website). The course parts on modeling cross the borders between the traditional engineering disciplines. They treat in a unified way mechanical, electrical, magnetic, fluid and thermal phenomena in engineering systems. This paper focuses on the course parts on control of these systems. The course is supported by the simulation package DYNAST accessible across the Internet. DYNAST can be applied to dynamic models of multidisciplinary systems characterized not only by equations or block diagrams, but also by configuration of real devices in the modeled systems. The course is structured so that the theoretical topics can be read in the textual material content but learned, and confidence gained in their use, by going through solved examples and open problems using DYNAST. Also facilities for linking DYNAST with virtual reality experiments across the Internet have been developed. These experiments have been incorporated into the course, with further work being done in associating the control design used in them to the course content. Several of these experiments do allow manual control so they can be used to illustrate what can be achieved with good control compared to manual operation. Thus in this context they can be used as excellent demonstrations to stimulate interest in control engineering.

2. SIMULATION IN DYNAST

2.1. DYNAST as A Distributed System.

The concept of DYNAST was laid out and implemented at the Czech Technical University in the early eighties. Since then, it has been developed further (mainly its graphical user interface) and well used both in academia and industry in a very wide range of applications. It can be utilized off-line as well as a software system distributed across the Internet. The former option allows for

- an easy access to a powerful simulation engine for students and engineers
- supporting cooperation of remote teams involved in common research or design projects
- giving distance education students a hands-on opportunity to solve real world problems

The fundamental parts of the distributed version are shown in Fig. 1.

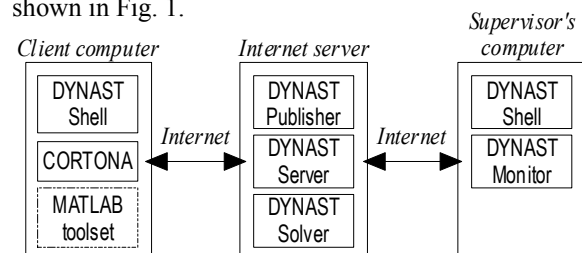


Fig 1. DYNAST distributed simulation system.

2.2 DYNAST Solver.

The kernel of the distributed simulation system is formed by DYNAST Solver which is capable of

- solving implicit sets of nonlinear algebraic-differential equations submitted in a natural textual form
- simulating nonlinear systems modeled by multipole and/or block diagrams submitted in a graphical form
- linearizing the diagrams and providing their semisymbolic analysis in the time- and frequency-domains

The most important advantage of multipole diagrams over block diagrams is in their similarity with the configuration of the modeled real system. The practical consequence of the similarity is that the multipole diagram of a system can be set up in a kit-like fashion from symbols of the system components in the same way in which a real system is assembled from real components. The symbols are interconnected by line segments representing real wires, pipes filled with a fluid, shafts or other mechanical links, and the bidirectional transfer of energy in them. Such a modeling procedure can be based on mere inspection of the modeled system. The underlying equations are then formed by DYNAST automatically based on physical laws governing the balance of energy, mass, or electric charge. Recollect that a block diagram is just a graphical representation of a set of equations usually formed manually. Each of the line segments interconnecting blocks is associated with just one mathematical variable propagating in one direction only.

The DYNAST Solver reads in input files with problem descriptions in a textual form representing equations and/or diagram netlists. The input data is directly interpreted, so there is no compilation delay as, for example, in Dymola. Besides transient responses, DYNAST also computes system steady states after converting the differential equations into algebraic ones. The algebraic equations can be solved also for a parameter swept through an interval. The transient responses can start either from initial conditions specified by the user, or from those corresponding to the system steady state. Even if the user-specified initial conditions are inconsistent, DYNAST is able to find the nearest consistent initial conditions within a few iterations. The fast Fourier transformation is available in this section for frequency-spectrum analysis of steady-state periodic solutions of nonlinear systems.

The DYNAST nonlinear equation-solving routine is based on a stiff-stable implicit multistep backward-differentiation integration formula. The length of integration steps and, at the same time, the order of the formula is optimized continuously to minimize the computation time while respecting the admissible computational error. Jacobians of the equations are

evaluated using a symbolic differentiation procedure. Considerable savings of computational time and memory are also achieved by exploiting the Jacobian sparsity. Thanks to the implicit form of the equations and to their simultaneous solving, DYNAST imposes no artificial restrictions on the simulated system structure. There is no need for equation sorting to solve the causality, algebraic loop and other problems associated with several other software packages, such as Simulink, for example.

After automatic linearization, the analyzed nonlinear system can be subjected to small-excitation analysis within the vicinity of a user-specified or computed quiescent operating point. Operator functions representing system transfer functions and transforms of initial-state responses can be then provided in a semisymbolic form with the Laplace complex variable s as a symbol and polynomial roots or coefficients as numbers. For such operator functions, DYNAST can compute semisymbolic-form time-domain characteristics. It also evaluates the corresponding characteristics numerically.

2.3 Working Environments.

The server-based DYNAST Solver can be accessed across the Internet in several ways (Mann and Sevchenko, 2003) Setting up the 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 computational results are sent back, they can be plotted on the client-computer screen.

An even more comfortable and user-friendly mode of access to DYNAST Solver is provided by DYNAST Shell, as shown in Fig. 1. This software can be freely downloaded and installed on client computers with MS Windows. DYNAST Shell has been designed for a wide variety of tasks in a way which is suitable for users of different levels of experience. All operations are supported by a context-sensitive help system. A built-in syntax analyzer continuously checks the submitted data. Unlike Modelica, or the IEEE Standard 1076 language, the DYNAST input language was designed as a natural engineering rather than a computer programming language. Yet DYNAST Shell dialogs (wizards) allow for the intuitive submitting of data without knowledge of the language.

DYNAST is accompanied by libraries of submodels for electronic and fluid-power devices, electro-mechanical transducers, mechanical parts, control units, etc. The submodel dynamics can be described by any combination of multipoles, blocks, or equations nested in a hierarchical way. 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. The libraries are open for easy addition of user-defined submodels and their symbols. Also Simulink like blocks have been easily added to DYNAST Shell and include graphical editors for multipole and block diagrams as well as for submodel symbols. A special dialog box for each new submodel is formed automatically.

DYNAST Shell can also communicate with the server-based DYNAST Publisher. It is a documentation system for automated publishing of reports on simulation experiments and descriptions of library submodels using LaTeX. The system extracts automatically the relevant parts of the input data and captures the submitted diagrams as well as the resulting output plots and includes them into the documents. The documents can be converted by the server into PostScript, PDF and HTML formats. Another very useful tool communicating with the server is the software package called DYNAST Monitor shown in Fig. 1. It allows design managers or tutors to observe from any site on the Internet the data files and diagrams the users are submitting to DYNAST Solver from their client computers. The supervisor can communicate with the users across the Internet and assist them in solving their problems.

3. CONTROL TEXTUAL MATERIAL

The control part of the textual material covers topics which are normally contained in a first course in control engineering. The material is included to provide basic background knowledge and reference material for the course coverage but it is hoped that the major learning experience for a course user will be obtained from 'doing', namely going through the suitably structured examples and problems provided in DYNAST.

The textual material on control is contained in Parts III and IV of the course. Part III, authored by C. Schmid, commences with an introductory chapter after which follow three modules covering the basic topics of classical control, namely: differential equation modeling, Laplace transforms, transfer functions, block diagrams, poles and zeros, step and frequency responses, steady state errors, stability and root locus plots. There are also modules on controller algorithmic design including optimization techniques, in more depth than might be covered in a first course, and detailed coverage of PID control including the classical methods of Ziegler-Nichols. There is also an introduction to state space methods and a chapter on fuzzy control. The latter makes clear to the reader that perhaps the name fuzzy is a bad choice in that a nonlinear controller with crisp inputs and outputs is obtained and the only 'fuzziness' is provided by the imprecise rules which

are used in the controller generation. A novel feature is a discussion of how changing membership functions and rules for a single-input single-output controller changes the shape of the nonlinear characteristic, which is of course a requirement for a physical understanding of why such a controller might be of value.

Part IV has been prepared by the first author of this paper. It deals with classical aspects of nonlinear control covering phase plane methods, the describing function, the exact determination of limit cycles in relay systems, and absolute stability. The emphasis on nonlinearity in Part IV is believed to be important in a course supported by simulation, because nonlinearity always exists in real problems, is not a constraint for simulation, and the methods presented can be helpful for interpreting simulation results. It is also stressed that if a system is non-linear then responses are amplitude dependent thus a step response for a single input amplitude, does not provide justification for the satisfactory behavior of a nonlinear system.

4. EXAMPLES AND PROBLEMS

The website for DYNAST has many examples covering several engineering disciplines including control. The examples are set up using a standard style sheet and those related to the course can be found under the control heading. Here just a few examples are discussed to show the range of coverage and to illustrate some important aspects which are sometimes neglected in control courses.

4.1 Steady State Errors.

Several examples are included in this category and a major emphasis is placed not on mathematical derivations using the final value theorem but on the physical understanding that an integrator is an active element which in the steady state can have a constant output with no input, or in other words it has infinite gain at zero frequency (d.c.). The example chosen can be found under the heading 'D.c. motor position control'. The purpose of the example is defined as 'To study closed loop position control using a DC motor and PID controller'. In the description of the problem it is stated that there are two inputs a reference position and a disturbance torque on the motor shaft. The user is asked to confirm and explain why for a reference input and no disturbance torque there is no position error if the controller only contains a proportional term but why an integral term must be included for zero steady state error when a disturbance torque exists. That there will be no steady state error to the reference input is generally well known to students (type 1 system etc., why do we teach it?) but for the disturbance they are expected to be able to understand that as the motor

contains an integrator it must have zero input for constant output, hence there must be integration in the controller so that it will produce a constant output equal and opposite to the disturbance for a zero error input. They are also asked to examine the reference step responses both without and with the integral term and to realize from this that there is a ‘penalty’ for the latter. This is a typical engineering trade off.

4.2 Relay Autotuning.

As mentioned earlier PID control is discussed at some length in the textual material and an example is given on this topic in the describing function section. The relay autotuning example has a general note on relay autotuning, pointing out additional facts about this method for estimating the critical gain and frequency. A few of these points are:

1. The results will be more accurate the nearer the limit cycle is to being sinusoidal.
2. Strictly the amplitude of the fundamental component of the limit cycle should be used for a in the formula $K_c = 4h/a\pi$, where $\pm h$ are the relay output levels. However, for ease of implementation half the peak to peak value is usually taken.
3. Other known networks can be placed in the loop to get information on other points of the plant’s frequency response or to make the limit cycle nearer sinusoidal to obtain more accurate estimates for the critical point.
4. When one does a simulation the plant transfer function is known so calculations can be done to calculate the critical point etc.
5. In a practical situation measurements will be contaminated by noise, so the accuracy with which measurements can be obtained may be worse than the approximation in the describing function analysis.
6. Measurements can be obtained more accurately, for the situation in 5, if averages are taken over several periods of the limit cycle, assuming the plant dynamics remains constant during the test.
7. It may only be possible in some industrial situations to have a limited time for the test, so that results for one, or even part of a limit cycle are all that can be obtained.

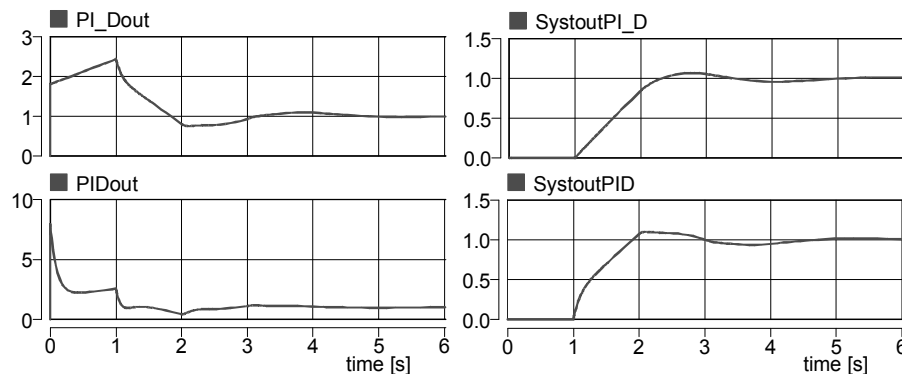


Fig. 3. PI-D Control – plots.

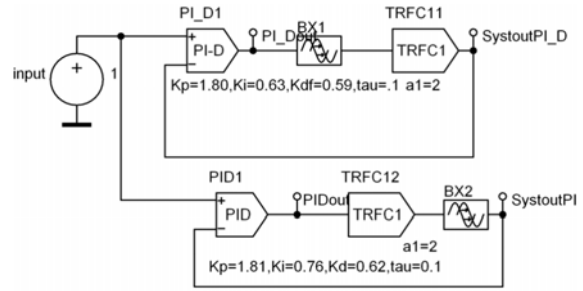


Fig. 2. PI-D control – model.

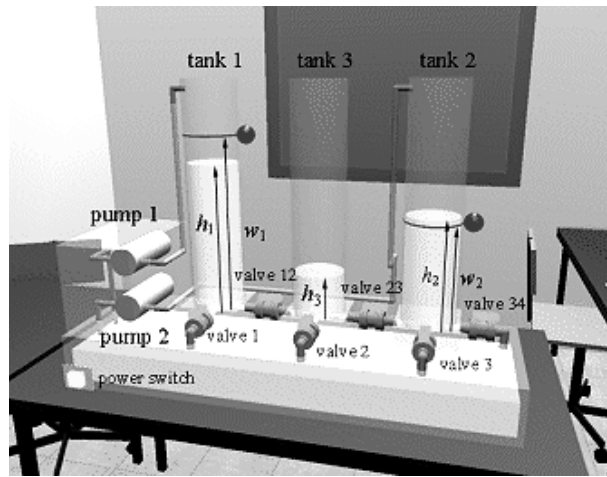
The example called relay autotuning has a third order transfer function and the user is asked to compare the critical gain and frequency values calculated from the measured limit cycle frequency and amplitude with the exact calculated values. Additional questions suggest further investigations with other plant transfer functions, comparing the accuracy of the results in relation to the limit cycle shape, examining the relative accuracy of the gain versus the frequency estimates, adding filters in the loop and so on.

4.3 PID Control.

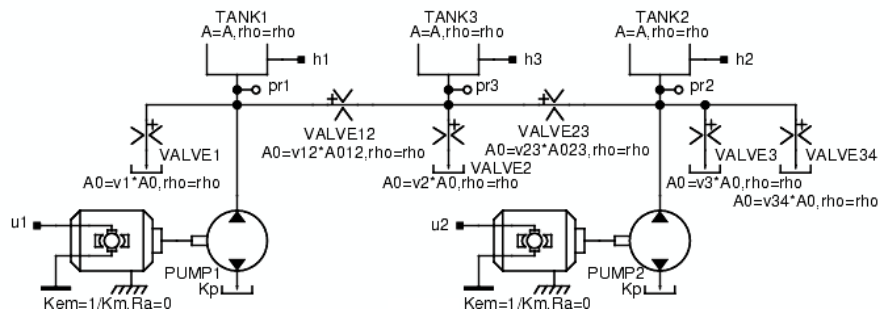
There are several examples set up to demonstrate PID control which again try to bring out aspects easily shown in simulation which textbooks often ignore. One of these is the effect of moving a compensator from the forward path, where it is usually considered to be in most textbooks, to the feedback path. It is rarely pointed out that this does not alter either the open loop frequency response or the closed loop poles but will alter the closed loop step response due to the change in zero positions. The PID control equivalent of this is to feed the D term from the output rather than the error, which will be denoted as PI-D control. To clearly demonstrate these effects one example, shown in Fig 2, is set up to run in parallel two simulations one of the plant, which is taken to be

$$G(s) = \frac{e^{-s}}{1 + 2s},$$

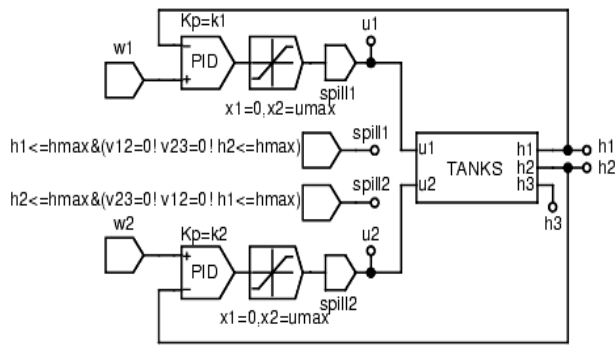
with PID control and the other with PI-D control. This is an easy way to compare different control strategies using simulation which seems to have received little exposure.



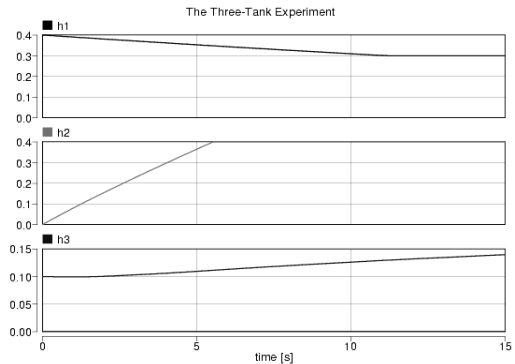
(a)



(b)



(c)



(d)

Fig. 4. Three tank virtual experiment.

As a starting point optimum parameters are given for minimization of the performance index

$$J_n = \int_0^{\infty} \{t^n e(t)\}^2 dt$$

for a step reference input for both control strategies with $n = 2$. The time constant of the filter with the derivative term can be varied. Importantly as well as the output responses the control signals are also displayed and the user is asked to examine them. In this way users learn about the derivative kick with PID control, how it is affected by the filter time constant, and that in practice designs often make it so large that the control signal will saturate and invalidate a linear analysis. Also it can be seen that it is avoided with PI-D control which can produce a comparable performance. The results for this

simulation for which the optimum PID and PI-D parameters for K_p , K_i and K_d are 1.81, 0.76, 0.62 and 1.80, 0.63, 0.59, respectively are shown in Fig 3, using a derivative filter time constant of 0.1. Further the collection of examples demonstrates that settings of the controller parameters can only be optimum if a precise performance index is defined and the choice of parameters minimizes it. The optimum parameters for some other choices of n in the above index are also given. In practice the problem is often identifying a suitable performance criterion for which there is a unique solution. A further example provides controller settings to minimise the above performance index for a PI controller for the above plant for (a) set point change and (b) a step disturbance input. These settings are of course different and the user should realize several points,

such as (a) an infinite number of parameter choices can be made by minimizing the same index but adding weights to the required contributions of the two inputs (b) with a modern controller one could select different parameters for set point change and disturbance rejection if benefits could be demonstrated and (c) and as a consequence of (a) there is really no point in just comparing responses when using different controller parameters for plants for which an analytical model is available unless the required performance is defined.

There are collections of problems on many topics, primarily based on the user being able to gain confidence in doing theoretical work, by checking the calculations by simulation. As explained in section 2 DYNAST has capabilities for doing frequency response plots, linearization and other features used in control investigations, but there is no facility for root locus plotting.

5. MODELING TOOLBOX FOR MATLAB

During the control design procedure DYNAST can act as a modeling toolbox for the MATLAB control-design toolsets installed on the client computer. The plant to be controlled can be easily modeled and the model validated using DYNAST. The transfer-function poles and zeros required for the control design can then be exported from DYNAST to MATLAB in an M-file. The plant model can then be augmented in DYNAST with the designed control structure and the performance verified. In the case of a digital control design, the resulting control structure is implemented in Simulink where the plant model, which remains in DYNAST, is represented by a block. Simulink installed on the client computer can then communicate with DYNAST at each time step across the Internet via an S-function.

6. VIRTUAL REALITY EXPERIMENTS

To stir up students' interest in dynamics and control as well as to enhance their understanding of the topics, the course text is augmented by 3D virtual experiments (Schmid, 1999), 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 students 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. 4b, Fig. 4c shows in detail the multipole model of the controlled plant, Fig. 4d gives transient responses of tank levels for a certain configuration of interactive elements in a similar way as if observed by measurement.

7. CONCLUSIONS

The paper has presented a brief overview of a web based course on control engineering developed to make use of the Dynast modelling and simulation package. The control content is one of many features within Dynast available on the web at <http://virtual.cvut.cz/dynlabcourse/>. 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 controller 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.

8. 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 provision of much of the textual material for the control part of the course and the original versions of most of the virtual reality experiments by C. Schmid, the Ruhr-Universität partner, which are mentioned in this paper, is greatly appreciated.

9. REFERENCES

- Website of the Leonardo da Vinci project DynLAB <http://virtual.cvut.cz/dynlab>.
- Mann, H., M. Ševčenko (2003). Internet-Based Collaboration and Learning Environment for Efficient Simulation or Control Design. *Proc. Congress ASME, DSC 71*, Washington D.C.
- Schmid, C. (1999). A remote laboratory using virtual reality on the Web. *Simulation*, **73**, 13-21.