A Web-based Design Tool for Control Systems

M. Gerke, A. Huber and I. Masár

Control Systems Engineering Group, Department of Electrical Engineering,
FernUniversität in Hagen, D-58084 Hagen, Germany
michael.gerke@fernuni-hagen.de

Abstract

This paper presents a Web-based on-line learning environment for the solution of control problems. The environment includes toolboxes for NYQUIST plots, BODE plots and for calculation of Root Locus contours. These frequency-domain tools are useful for control systems analysis and synthesis with respect to overall system stability and dynamic performance of closed-loop circuits.

It is available on the Web especially for students, trainees and home-learners in the field of control theory.

1. Introduction

The concept of a Virtual University has been arisen almost a decade ago. Starting with simple communication tools such as ‘Email’ and ‘Newsgroups’, this concept has been enhanced with many different ideas and applications. Learning platforms as a framework for Internet universities have been developed providing various helpful tools and instruments to build a consistent system of information sharing and electronic courseware distribution. Content management systems have been integrated to enable a user-friendly input of information. Computer supported collaboration re-introduces groupwork into a Virtual University environment. Remote laboratory infrastructures provide access to Internet-based experiments especially for students in engineering, natural and computer sciences. Virtual seminars can be carried out via video and audio conferencing on a personal computer. Virtual Reality techniques are also applied to many different learning scenarios.

One of the most important features to support Web-based training is the visualization and animation of examples given in a courseware context. This is an important add-on functionality, which is rarely available for students of conventional universities. Even simulations can be included into electronic courseware to motivate students to explore further on and to have a look behind the scenes. In this paper we focus on Web-based tools to enhance learning and training in the field of control theory.

Nowadays advanced Internet techniques provide powerful tools for client-side applications such as complex simulation environments for control problems. Based on the Java Virtual Machine (JVM) which is available for almost any operation system and hardware architecture, Java Applets included into Web pages can perform various tasks on the host computer system. In our case, we programmed applets for formula parsing (input of transfer functions) and control system analysis and synthesis (calculation and graphical representation of plots).

The design of a user input interface for describing the dynamic systems behavior via mathematical formulas and the subsequent formula parsing is discussed in section 2. The calculation of NYQUIST plots (section 3), BODE plots (section 4), and Root Locus contours (section 5) as well as their graphical visualization is outlined in the sections following next.

The benefit of Web-based tools in the context of Virtual Universities is briefly discussed in section 6. Some remarks and comments conclude this paper.

2. The User Interface: ‘Start Applet’

The calculation of plots and graphs is based on proper input of open-loop transfer functions. To keep this data input simple, the user of our Web-based tool can choose between two options: polynomial input or input in MATLAB® notation.

The polynomial input requires a nominator and a denominator polynomial in ‘s’. As a condition to simplify formula parsing, the coefficient ‘a0’ always has to be explicitly used, even if it is zero. An example for polynomial input is given in figure 1.

The convenient MATLAB notation uses square brackets to include the coefficients of a polynomial in sorted order. Nominator brackets come first. An example for MATLAB notation is presented in figure 2.

Formula parsing is implemented with some methods of string manipulation, that are available in Java class libraries [Wi02],[Da02].

Regular expressions are used to distinguish between both input options and to split input expressions into variables, that can be processed within the control tools. The class ‘PolyParse’ as a part of the ‘hmath’-package is included to verify input data according to the predefined conventions and to determine the order of polynomials.
For purpose of graphical presentation, a ‘Java Script’ is used to set the resolution, width and height of all applets by analyzing user screen data.

The ‘hmath’-package also provides the HORNER algorithm, which is very efficient to calculate the functional value ‘F(s)’ of a polynomial with given coefficients $a_0$ to $a_n$ and for any given ‘s’. This algorithm is implemented into our applets for calculation of system dynamics. Finally, the ‘Start Applet’ offers different options for further control considerations. The three buttons ‘NYQUIST plot’ (in German: ‘Ortskurve’), ‘BODE plot’ (in German: Frequenzkennlinie) and ‘Root Locus’ (in German: Wurzelortskurve) lead to the different control design tools of this Web-based environment.

3. NYQUIST plots

Beneath the elementary task of calculating single points of the NYQUIST plot in a complex z-plane, it is the more challenging task here to derive closed curves from these points and to consider an appropriate scale factor, that leads to a plot presentation, that uses the full size of the graphical output applet. An example of a successful implementation is given in figure 3.

Whatever the result of a NYQUIST calculation is, the user can scale up/down the plot area with the ‘Zoom’-function thus multiplying the real value ‘Re(z)’ and the imaginary value ‘Im(z)’ with a scale factor.

**Figure 1: Polynomial input of a transfer function**

**Figure 2: MATLAB input of the same transfer function**

**Figure 3: NYQUIST plot of input function**

JAVA provides appropriate functions for these scaling operations such as the casting operator ‘int’ and the floating point operator ‘double’.

Some more considerations have to be made considering limitations and boundary values of pixel coordinates.
However, the mathematical methods in Java libraries are very 'programmer-friendly', because they do not deliver irregular values but exceptions. This fact is used intensively to resize the numerical range of the applet window during calculation of single pixel values. Also a maximum step width and a frequency range for calculation have to be defined and monitored; a break condition is implemented to avoid convergence problems.

4. BODE plots

The problems to be solved for BODE plots are quiet comparable to those to be considered in context with NYQUIST plots. For graphical presentation, it is an enormous advantage here, that functions have to be plotted with respect to the frequency axis 'log(ω)'. A grid underlies the plot in order to simplify the allocation of important curve features. Again, an example is given in figure 4. It is subdivided into amplitude and phase plot.

![BODE plot of input function](image)

Some additional user features have to be mentioned:

The ω-range of the frequency plots can be altered with the horizontal scroll bar. Amplitude and phase range are also subject to user modification through activation of the plot window (mouse located in window signals an event) and then alteration of view range with the arrow keys. It should be mentioned, that any alteration of the ω-range requires a re-load of the whole 'html'-page.

5. Root locus contours

Calculation of root locus contours 'by hand' with the help of design rules is an unpleasant and time-consuming task. Usually computer-based calculation does not follow the design rules, but carries out an exhaustive search in the range of interest (e.g. applet window). In a first approach this could be done for all point in the applet window. The idea is to find out, whether a point meets the root locus condition or not.

The search space can be reduced drastically, if the following assumptions are made:

- Root locus contours always start in the poles of the open-loop F0(s).
- Often the contours follow the real axis of the complex plane.
- With parameter factor ‘k’ increasing, the contour points move steadily through the plane.

Due to the non-linear behavior of contour points with respect to ‘Δk’, the problem is to find an appropriate variation of ‘k’ without going lost [AK70] in search space. On the other hand, in general there is no need for an exact presentation of contours according to their modulus and phase condition.

In [AK70] an 'unsharp' Root Locus condition is formulated, which make the algorithmic solution of the search problem more feasible. Briefly, it is based on the search of ‘zeros’ for the characteristic systems equation of F(s) = 0.

Once again, the HORNER class of Java is extremely helpful for the algorithmic implementation of the Root Locus contour search. Some basic considerations also simplify the location of ‘zeros’ of F(s) in a search grid by means of analysis of boundary values in any square of the grid.

The Root Locus tool also offers some helpful user features: a ‘Zoom’ function simplifies the analysis of contour details, which are often quiet surprising because of their non-linear dependency on Root Locus parameter ‘k’. Also the mouse pointer allows a precise assignment of real value and imaginary value positions and parameter values of ‘k’ for any point of the contour.
Again (as in all other applet tools), the plot can be move horizontally and vertically with left mouse button being pressed down.

Figure 5 gives an impression of a root locus contour for the unstable transfer system under consideration throughout this paper.

Figure 5: Root locus plot of an unstable control-loop

6. Conclusions and Remarks

Web-based tools are an important learning add-on and a benefit not only for trainees and students of ‘Virtual Universities’, but also for students of conventional universities and schools. All these people have to learn at home in order to get a deeper insight and a more practical understanding of the usually more theoretically oriented lessons taught in classrooms (or provided by electronic courseware). Some of these learners use such tools for exam preparation. But also practitioners have a benefit from Web-based tools, that address their field of expertise and probably these tools can support them during problem solving.

To focus on control theory and controller design, it is quiet obvious to understand, that some sort of experimental environment is urgently necessary for students to find out and explore what happens, if system dynamics change or if controller parameters are altered. This can only be learnt by doing.

It is due to this fact, that tools like ours are under development in many universities. Most times, these tools are distributed with commercial books or offered to a restricted user group only.

Still often, these design tools are accessible via Web-pages of university institutes, but they remain unrelated to the teaching program of the faculty. However, sometimes these tools are really integrated into learning platforms, so-called ‘Virtual University’ environments. Here they are in close conjunction with the courseware itself, e.g. via links from examples introduced in the theoretical fundamentals of the course. Libraries of solved examples can enhance these environments.

In our case, the Web-based design tool has become a mosaic of our ‘Virtual University’ (FernUniversität in Hagen is the only German university based on the principles of distance education). Moreover, we have implemented a consequent learning environment for control systems teaching:

1. Electronic courseware to introduce the theoretical fundamentals of control theory.
2. A closely related system of exercises to be solved ‘by hand’ in order to learn the basics of problem solving in control systems engineering.
3. A Web-based design tool (presented here) for control problems to become familiar with computer-based problem solving. Interfaces to MATLAB/Simulink® are available.
4. An Internet-based ‘remote lab’ for carrying out real experiments in our lab from the user host computer at home.

We believe that all these different methods and tools are necessary to establish an environment for the education of control systems engineering students in a ‘Virtual University’.

9. References