

MATLAB/Simulink Based Restructuring of the First Automatic Control Course in Engineering Curricula

Mehmet Önder Efe

Hacettepe University

Autonomous Systems Laboratory, Department of Computer Engineering

Ankara, Turkey

onderefe@hacettepe.edu.tr

Abstract—Most engineering schools offer the first Automatic Control course in 6th or 7th semester of the curriculum. The students taking this course have already a background on linear algebra, differential equations and signals & systems. The first Automatic Control course used to cover the same topics yet the style of instruction in this paper is a Matlab based one. The strengths, weaknesses, opportunities and threats are discussed in the paper.

Keywords—Blended learning, Use of Matlab, Control Systems Course

I. INTRODUCTION

Automatic Control course is offered in a number of disciplines, Electrical & Electronics Eng., Mechanical Eng., Computer Eng. And Chemical Eng. Curricula are those including this course. Typical prerequisites for this course are the Linear Algebra and Differential Equations. Some disciplines further require Signals and Systems.

With this background, a typical catalog data for a 6th or 7th semester Automatic Control course contain: Laplace Transform, Stability Check with Routh-Hurwitz Table, Transient Analysis, Steady State Analysis, Frequency Domain Analysis, Bode Plots and Stability Margins, Root Locus Technique, Nyquist Criterion, Design of Phase Lead/Lag Compensators, PID Control, State Space Analysis of Control Systems, Controllability, Observability, Pole Placement and Observer Based Design.

Looking at the items listed above, one sees that each one requires a particular knowledge of mathematics, algebra and differential equations. Old fashioned teaching of these subjects has frequently failed in relating the results with other items. For example, changing the shape of the root locus changes the Bode phase and magnitude plots, transient response, Nyquist plot, step response and the like. In classical teaching approaches, it is difficult to show these for a number of exemplar cases as the calculations for each case is tedious and it is difficult to maintain the interest of the students. The use of new generation approaches have blended the computer aided design tools with higher education strategies and the learning experience of the students have greatly been improved.

Matlab is a powerful tool for engineering students and the Automatic Control courses have benefited from the computational facilities provided by this package. Especially Simulink has been a core design alternative as it is based on graphical components. In the literature, a number of studies considered the use of Matlab for educational purposes.

In [1], a control laboratory example is discussed. The target group is undergraduate students and sharing of resources and standardization of the course is emphasized. Guzman et al. discuss the role of interactive tools in teaching control courses, [2]. A low cost solution to control systems laboratory is proposed in [3], where the authors utilize a microcontroller based control of a DC motor. The didactic value of this kind of an approach is still a good solution adopted in several institutions. In [4], differential relay protection using Matlab/Simulink is studied, and in [5], Youla parameterization is studied as a new control paradigm in basic control education. Zakova, [6], presents a web based solution for control education. The proposed approach utilizes Matlab as the numerical platform. Ćirka et al, [7], demonstrate the design of PID controllers using Matlab/Simulink and Saxena et al present a GUI that is used for performance analysis of a control system, [8]. In [9], an interactive PID learning module is discussed. The proposed module is developed for educational purposes. In [10], the Matlab based teaching approach is utilized for the remote control of a 6 degrees of freedom robotic system.

Clearly the use of Matlab/Simulink as an interactive tool for teaching is not a new topic yet the profile of the incoming students is changing and an optimum teaching strategy needs a careful reorganization of the offered courses.

This paper is organized as follows: The second section discusses the strengths, opportunities, weaknesses and threats of using a Matlab based approach to teaching the first Automatic Control course. The third section considers the major subtitles of the course and possible roles of Matlab and/or Simulink in each subject. In the fourth section, an example is considered and its role as an auxiliary tool is discussed. The concluding remarks are given at the end of the paper.

II. STRENGTHS, OPPORTUNITIES, WEAKNESSES AND THREATS

The use of Matlab for teaching the Automatic Control course can have the following impacts on the overall teaching performance.

A. Strengths

- Use of CAD software like Matlab/Simulink helps developing a concrete dynamic model, which can be analyzed in detail.
- Students can run the model under different operating conditions and quickly obtain the response and interpret it.
- Quick experiments with DSP utilities can be brought to a theoretical session of the course and more didactic teaching atmosphere can be established.
- Standardization of the material used for large groups can be obtained by the use of CAD based solutions.
- Sharing of resources can be obtained in courses that utilize the possibilities of CAD software like Matlab/Simulink.

B. Opportunities

- Students can obtain a prototype quickly and this improves their motivation.
- The use of acquired knowledge can be utilized in interdisciplinary design problems.
- Matlab/Simulink quickly becomes a design language that enhances the quality of information exchange among the mates.
- Interactive learning possibilities can be enhanced.
- The taught concepts become permanent throughout the professional life of the candidate engineer.

C. Weaknesses

- The approach may easily encourage the feeling of “learning the underlying mathematics and physics is unnecessary”
- Web based laboratories are very popular and Matlab/Simulink oriented experimentation experience makes the young engineers feel unsatisfactory in real operation fields.
- Software centric teaching requires the redesign of the exams, projects and home works. The grading of these activities will become harder when the software part constitutes the core of the design.

D. Threats

- Students may adopt the “trial and error” as a design approach
- The teaching of Automatic Control courses are gradually being more blended to Matlab/Simulink based analysis and the jargon is quickly changing. This is considered as a threat in the category of engineering communication.
- The use of CAD software can easily become the first step in design, which may influence the creativity adversely.

III. A RESTRUCTURED AUTOMATIC CONTROL COURSE

The approach discussed here augments the theoretical part of the instruction by the following Matlab/Simulink modules.

Laplace Transform: The transfer functions and their inverses (the impulse function) are shown on Simulink diagrams. The use of Matlab/Simulink, changing the simulation settings and obtaining an acceptable result, effects of the simulation step size are shown through interactive examples.

Stability Check with Routh-Hurwitz Table: Especially the cases containing a zero in the first column and a row with all zeros are studied with Matlab/Simulink examples. These contain the analysis of impulse and step responses.

Transient Analysis, Steady State Analysis: The built-in command “`sisotool`” in Matlab is a useful resource for teaching. The change of the locations of the poles of a transfer function has an impact on the transient and steady state response and seeing this in an interactive session is found very helpful.

Frequency Domain Analysis: The Bode magnitude and phase plots are taught with several primitive cases, more complex cases are taught via asymptotes. With this in the front, students still need a visual example that shown how the Bode plot changes if the gain, poles and zeros change. Use of Matlab for this purpose is a very helpful and illustrative experience. The notion of gain and phase margin is another issue that needs a CAD support for effective learning. The locations of the poles and zeros can be changed and the value of the system gain can be increased, all these have effects on the relative stability of the system and showing these through an interactive session results in a better understanding of the notion.

Root Locus Technique: One of the most abstract topics in the first Automatic Control course is root locus technique. Simpler cases can be understood quickly yet an increase in the number of poles and zeros result in a more complicated root locus picture and interpreting the response of the system as well as its stability properties can be taught very effectively using the “`sisotool`” command of Matlab.

Nyquist Plot and Nyquist Stability Criterion: Use of a tool that changes the Nyquist plot instantly is helpful resource as the importance of the deployment around the point $-1+j0$ and encirclements of it change the stability of the system.

Design of Phase Lead/Lag Compensators: This title is the first design stage that proposes the changes in the closed loop and expects the desired transient and steady state responses. The change of stability and its margins become visible with an interactively result updating CAD tool and the student grasp the design process well. Specifically, the effect of lag action on the steady state performance and the effect of lead action on the transient phase can be studied with examples.

PID Control: When the model of the plant is known the performance of the PID controller is good. This can be shown by an example and the model can be changed and the degradation of the PID performance can be shown interactively. Matlab/Simulink has several PID Controller modules that offer an auto tuning scheme. The results can further be optimized to observe a fine tuned response. This tool should be considered as a complementary tool of theoretical

sessions focusing on Ziegler-Nichols and similar tuning schemes.

State Space Analysis of Control Systems: Matlab offers a very powerful set of tools for analysis and design of control systems that include state space models. These include the checking of controllability, observability, pole placement and observer design. Further to these, a number of built-in commands provide conversions from state space to transfer function, or the reverse.

With this picture in the front, a restructured Automatic Control course can be implemented with an active use of Matlab/Simulink to demonstrate the effects of any parameter that takes place in the design process. The graphical user interfaces illustrate the fundamental properties of the control system together and instant changes gives a clear idea about the design issues.

At this point, the instructor may consider how to customize the course for students from different disciplines. This is a critical issue e.g. for Computer Engineering students taking the Automatic Control course. The discipline tends to Computer Science in some institutions and the analytical background of the students become insufficient to handle the mathematics of the coursework. In these cases, the role of Matlab/Simulink based teaching becomes even more suitable to convey the design issues appropriately.

A final note in this section could be the use of simple hardware during the lectures. Matlab/Simulink has a number of interfaces that comfortably connect to any Field Programmable Gate Array (FPGA) board or simpler cards (like Arduino, Raspberry or the like) used for hobby purposes. Use of these motivates the students significantly and the courses could claim a final project in some institutions.

IV. AN EXEMPLAR CASE FOR BETTER LEARNING

A dynamic system model is given as in (1), where the poles of the model are at the origin and at $s = -0.01$, which is very close to origin. Here, s denotes the Laplace variable and the system connectivity is a standard unity negative feedback one.

$$G(s) = \frac{400}{s(s+0.01)} \quad (1)$$

The transfer function G can be defined as below:
`>> G=tf(400,[1 0.01 0]); sisotool(G)`. The “sisotool” command will open the window shown in Fig. 1, where the connectivity and the transfer functions are defined. In Fig. 2, without any modifications, the root locus plot and the open loop Bode plots are provided. One should not that the student can change the gain using the GUI and can observe the effects of the changes instantly on Fig. 3, where several responses/plots of the system are shown. Since the compensator is unity ($C(s)=1$), the plot describing it has 0 dB magnitude and 0 radians phase at all frequencies (See bottom left subplot of Fig. 3). This means that the loop is uncompensated.

The second stage of the learning experience is to observe what happens if a new real pole is added. The students see how the stability properties change when the location of the pole is changed. After this, a real zero is added and the same process is continued.

The next phase of the design is to compensate the loop appropriately. Some design specifications are imposed: For the particular case discussed in this paper, the students are asked to cancel the real pole of the plant and add a new pole. This has yielded the following open loop transfer function;

$$G(s)C(s) = \frac{400K}{s(s+a)} \quad (2)$$

where, a and K are the pole location and the open loop gain to be determined according to the design specifications.

At this stage, students are asked to set $a = 2$, and set the gain K so that the response to a step command does not exhibit overshoot and oscillations. This gives the following compensator;

$$C(s) = 0.0025201 \frac{s+0.01}{s+2} \quad (2)$$

Note that the chosen gain is very close to $1/400$, inverse of the open loop transfer function. The figures obtained for the obtained compensator are shown in Figs. 4-5, where the changes can be seen comparatively with Figs. 2-3. The locations of the chosen zero and pole are shown as well as the gain margin and phase margin are also indicated on the resulting figure.

The question at this point is whether this helps the students or not. Clearly an Automatic Control course covers a number of major subtitles, each of which take about two weeks of the classical 14-weeks semester. This basically means that at the end of the semester, the instructor must explain the links in between the subtitles listed in Section III and the possibilities offered by Matlab and Simulink are invaluable in that sense.

Matlab has recently introduced a PID tuner to get the best performing PID controller both for continuous time and for discrete time system models. This significantly facilitates the understanding of Ziegler-Nichols based tuning mechanism and further refinement process. In [11], a number of different design cases with several design constraints are considered via exemplar cases.

`pidTuner` command provides a number of design options to the user. For the example considered one can easily call the design module by typing `>> pidTuner(G)`. The GUI provides tuning for all possible combinations of P, I, D and derivative divisor options. The response can be made slow or fast; the transient behavior can be made aggressive or robust, depending on the design specifications. The GUI also provides a number of plots to analyze the behavior of the closed loop system. An exemplar case is shown in Fig. 6, where the controller is a PID with a derivative divisor. The designer can work in the time domain as well as in frequency domain. In the latter, the bandwidth and phase margin are adjusted manually and the response of the closed loop is updated instantly.

V. CONCLUSIONS

This paper discusses the teaching of the first Automatic Control course to students and the use of Matlab/Simulink as a tool to facilitate learning. A SWOT analysis is presented and the needs of the contemporary learning strategies are discussed. The student is the core of the entire process and the student profile is changing as the computing and communication infrastructure changes. Higher speed, longer availability, quick communication, fast computers influence the student and the classroom. These have some impact of the overall performance of the instructor. Automatic Control is a particular course that is offered by a number of different disciplines. The profile of students taking this course is even broader than the number of disciplines and this motivates us to develop an enhanced teaching medium. Use of Matlab/Simulink during the theoretical part of the lectures is an approach that entails the redefinition of the semester-long activities according to this choice.

VI. ACKNOWLEDGEMENTS

The author gratefully acknowledges the facilities of the Computer Engineering Department of Hacettepe University.

VII. REFERENCES

- [1] W.E. Dixon, D.M. Dawson, B.T. Costic, M.S. de Queiroz, "A MATLAB-Based Control Systems Laboratory Experience for Undergraduate Students: Toward Standardization and Shared Resources," *IEEE Transactions on Education*, vol.45, no. 3, pp.218-226, August 2002.
- [2] J.L. Guzman, R. Costa-Castello, S. Dormido, M. Berenguel, "An Interactivity-Based Methodology to Support Control Education: How to Teach and Learn Using Simple Interactive Tools," *IEEE Control Systems Magazine*, pp.63-76, February 2016.
- [3] M. Gunasekaran and R. Potluri, "Low-Cost Undergraduate Control Systems Experiments Using Microcontroller-Based Control of a DC Motor," *IEEE Transactions on Education*, vol. 55, no. 4, pp.508-, November 2012.
- [4] B. Vahidi, E. Esmaeli, "MATLAB-SIMULINK-Based Simulation for Digital Differential Relay Protection of Power Transformer for Educational Purpose," *Computer Applications in Engineering Education*, pp.475-483, 2013.
- [5] R. Bars and L. Keviczky, *Introducing New Control Paradigms in Basic Control Education - YOULA Parameterization*, Szechenyi University Press, pp.107-112, 2015.
- [6] K. Zakova, "Web-Based Control Education in Matlab," in *Web Based Control and Robotics Education* (Ed.) S.G. Tzafestas, pp.83-102, Springer, 2009.
- [7] L. Čirka, M. Kalúz, J. Oravec, and N. Míková, "Designing PID Controllers using Matlab-Simulink via the Internet," *22nd Annual Conference Proceedings of the Technical Computing, Bratislava*. Vol. 22. 2014.
- [8] S. Saxena, A. Palak and Y.V. Hote, "User-friendly MATLAB based graphical user interface for performance analysis of control system," *IEEE International Conference on Computer and Computational Sciences (ICCCS)*, pp. 26-31, Jan. 27, 2015.
- [9] A. Theorin and C. Johnsson, "An interactive PID learning module for educational purposes," *19th IFAC World Congress*, August 24-29, Cape Town, South Africa, 2014.
- [10] M. Ruderman, F. Hoffmann, and T. Bertram, "A Matlab-based framework for the remote control of a 6DOF robotic arm for education and research in control theory," *Advances in Control Education*, vol. 9. no. 1., pp.366-371, 2012.
- [11] <http://www.mathworks.com/help/control/ref/pidtune.html>

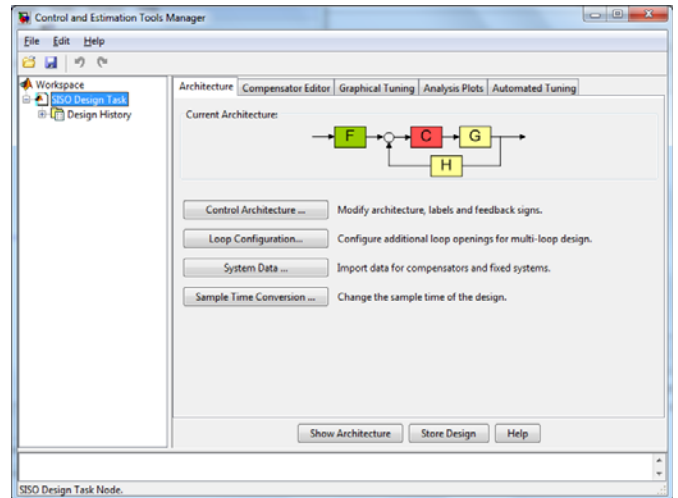


Fig.1. The window describing the connectivity and providing the main control tabs.

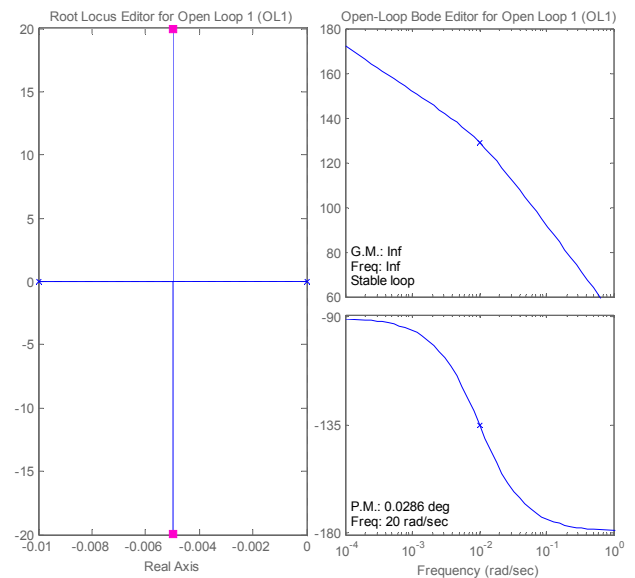


Fig.2. The window showing the root locus (left), open loop Bode magnitude plot (top right) and open loop Bode phase plot (bottom right). The loop is uncompensated.

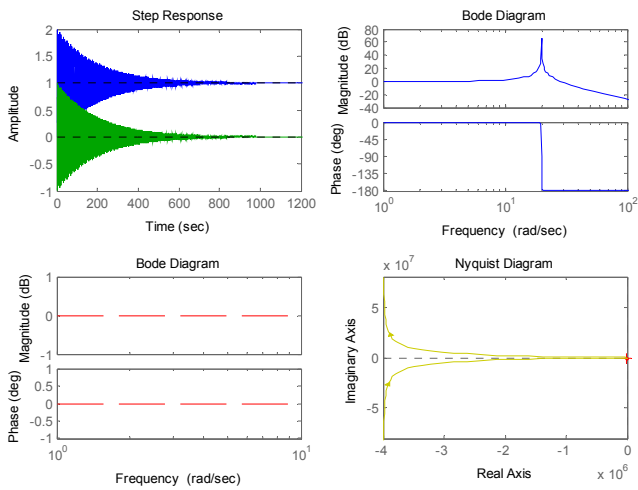


Fig. 3. Step response (top left), closed loop Bode diagram (top right), compensator's Bode diagram (bottom left) and the open loop Nyquist diagram (bottom right). The loop is uncompensated.

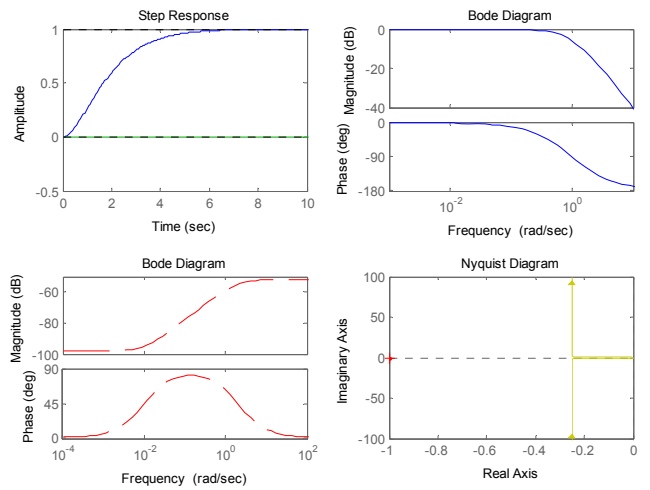


Fig. 5. Step response (top left), closed loop Bode diagram (top right), compensator's Bode diagram (bottom left) and the open loop Nyquist diagram (bottom right). The loop is compensated.

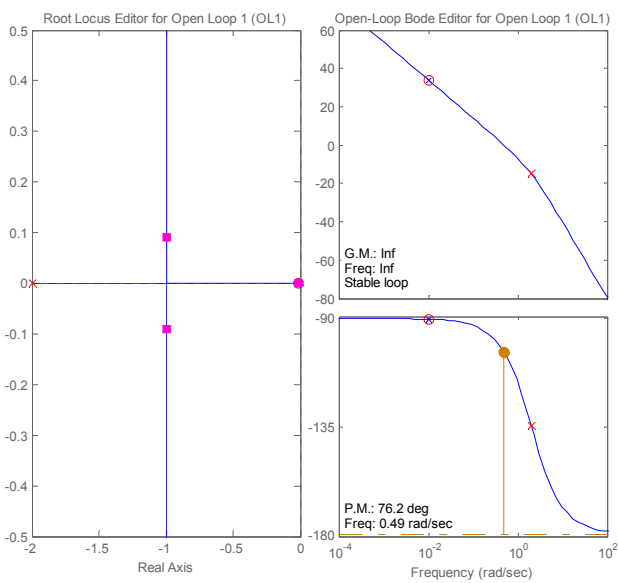


Fig.4. The window showing the root locus (left), open loop Bode magnitude plot (top right) and open loop Bode phase plot (bottom right). The loop is compensated.

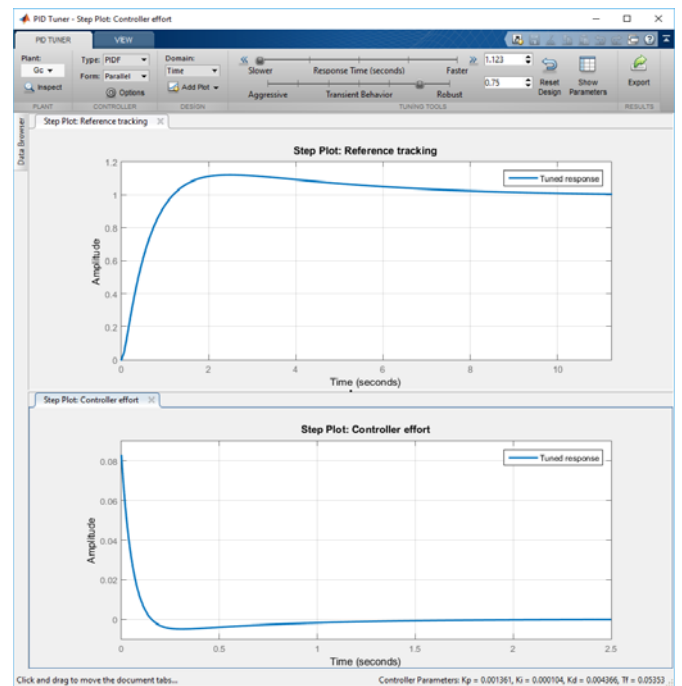


Fig. 6. An illustrative use of pidTuner command.