
Proceedings Paper:
Rossiter, J.A. orcid.org/0000-0002-1336-0633 (2024) MATLAB files to support learning of simple frequency response design. In: Caramihai, S. and Cieslak, J., (eds.) IFAC-PapersOnLine. 4th IFAC Workshop on Internet Based Control Education - IBCE 2024, 18-20 Sep 2024, Ghent, Belgium. Elsevier BV , pp. 1-6.

https://doi.org/10.1016/j.ifacol.2024.10.261

MATLAB files to support learning of simple frequency response design

J. A. Rossiter *

* *Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK (e-mail:j.a.rossiter@sheffield.ac.uk*

Abstract: This paper presents a small number of MATLAB APPs and livescript files designed to help students both understand and implement frequency response tools into feedback design. The paper presents the thinking behind the use of MATLAB and the topic itself before then describing the proposed resources in detail.

Keywords: Control101 toolbox, frequency response, lead and lag compensation, virtual laboratories, independent learning.

1. INTRODUCTION

One area of increasing interest in the control community and a focus for the IFAC technical committee on control education is the effective production and sharing of control education resources (Rossiter et al., 2018; Serbezov et al., 2022). The internet facilities rapid communication and sharing and thus it is no longer necessary for each *teacher* to produce their own resources. Instead it is possible to make use of resources produced elsewhere by careful integration into your own delivery (Douglas, 2022).

The author of this paper has recently proposed a specific contribution to these shared global resources, that is a MATLAB toolbox (Rossiter, 2023, 2021) focused primarily on a first course in control (Rossiter et al., 2020). The hope is that the control101 toolbox fits neatly into the delivery of a first course in control as it meets one of the fundamental aims of such a course which is to immunise learners, as much as possible, from tedious pen and paper computations so that they can focus on learning and appreciating core concepts and principles: for example, what is feedback and why is it important? While there are a number of excellent text book resources already in existence (e.g. Dorf et al. (2021); Golnaraghi et al. (2017); Ogata (2009); Nise (2017)), as a rule these focus on paper and pen computations and core background whereas the resources discussed in this paper, focus very much on using the computer to support learning.

The control101 toolbox is made up of two types of resource:

(1) Livescript files: These combine MATLAB coding with a neat presentation of the underlying engineering and thus support more traditional notes (eg. PDF files) by providing an interactive environment in which students can quickly and transparently explore the impact of changing parameters on the solution of set scenarios (without having to do the computations manually). Livescripts support figures and numbers and thus the solutions can be shown in a helpful manner (Rossiter, 2023).
(2) Virtual laboratories or *apps*: MATLAB has an interactive environment where the interface is essential a GUI (graphical user interface) so students press, buttons, sliders, drop down menus and so forth and the results are presenting in figures and text as appropriate. In essence this can serve as a virtual laboratory environment (Rossiter, 2022).

A first draft of the control101 toolbox was released at the IFAC world congress in 2023 and the intention there was to try and galvanise some interest from international colleagues so that ultimately the toolbox comprises a range of contributions representing the interests and needs of the global community.

Remark 1. Users can download the toolbox for free by searching for "control101" under *add-ons* in MATLAB or via github (Rossiter, 2023). The website also contains links to short youtube vidoes which are designed to help potential users understand how to use the files and virtual laboratories.

Virtual laboratories have been a growing area of interest in the community for several years (Matisak et al., 2023; Cameron, 2009; de la Torre et al., 2013, 2020) and there is significant evidence (Rossiter et al., 2018; Rossiter, 2017) therefore that these are effective learning tools. Nevertheless, in the context of the toolbox, this paper takes a slightly different direction in that it focuses on virtual laboratories that are not directly linked to any specific hardware. We want some interactive GUI resources Guzman et al. (2018, 2023) which allow students to engage with key concepts and principles so that they can learn and understand them better.

More specifically, this paper focuses on an introduction to frequency response methods, that is Bode diagrams, gain and phase margins, and simple compensator design approaches using these tools. Such topics may be at the end of a 1st course in control or the beginning of a 2nd course in control and thus are a reasonable fit with the

Control101 toolbox. Section 2 gives a quick review of frequency response methods and the core topics to be considered and thus identifies the requirements for the toolbox. Section 3 then introduces the files. The paper finishes with some reflections and conclusions.

2. AN INTRODUCTION TO FREQUENCY RESPONSE METHODS

This paper assumes that most readers are familiar with the basics of frequency response and hence the purpose of this section is more to lay out the prioritised topics rather than to introduce something new. It is also worth mentioning, that in the spirit of a 1st course in control Rossiter et al. (2020) the aim is not to dwell to much on first principles, derivations and hand sketching but rather on interpretation and use. The current apps and livescript files focus on the concepts/content listed below, although the authors are happy to accept this list could be extended and indeed would welcome further contributions from colleagues. The most up to date content listing is available by download Rossiter (2023).

(1) What is frequency response in the time domain?
(2) Capturing frequency response data using Bode diagrams.
(3) Links between closed-loop behaviour and open-loop Bode diagrams and gain/phase margins.
(4) Proportional design using phase margins.
(5) Lag compensator design using phase margins and low frequency gain.
(6) Lead compensator design using bandwidth and phase margins.
(7) Lead-lag compensator design.
(8) PID design using frequency response methods (work in progress).

The following section will introduce each set of resources in turn. In general there are both livescripts and virtual laboratories for each theme and these serve as mutually supporting resources. The detailed explanation of the design steps is not included here, but available in the livescript files and other standard resources.

Remark 2. It is noted that the MATLAB environment *sisotool* does encompass much of the functionality listed above, but the decisions, system and compensator definitions and so forth are neither as transparent nor convenient from a teaching perspective and thus this tool is harder to use in a piecemeal fashion to learn and illustrate one concept at a time.

3. THE CONTROL101 TOOLBOX LIVESCRIPT AND APP FILES SUPPORTING FREQUENCY RESPONSE METHODS

3.1 Files used in the resources

Livescript files serve the dual purpose of providing background information and explanation alongside suitable MATLAB coding. Consequently, these are flexible enough for students to use and edit the embedded code to investigate their own examples and thus to personalise as required. By providing code in this format, it minimises the need for students to understand detailed coding and instead they can focus on the engineering while MATLAB performs the background calculations and plotting to support their learning. For example, with many code snippets it is enough to edit the core system definition and requirements information in the top few lines while the remainder of the code need not be touched.

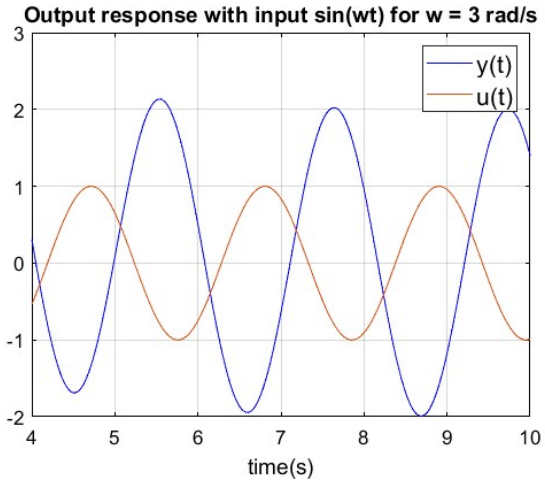

Fig. 1. Illustration of frequency response by comparing input and output behaviour.

The app files hide the code from students and allow them to focus solely on insights, parameter choices, illustrations and so forth. These are an ideal support to livescript files and indeed a lecturer may choose to use the apps first to motivate and interest students and later encourage them to use the livescripts in order to perform more detailed and individualised investigations. Each app comes with a manual which gives more detailed explanation of how to use and understand the interface; the manual opens automatically when the app is run.

3.2 What is frequency response in the time domain?

Frequency response is not a characteristic or definition that comes naturally to everyday life and thus some time needs to be spent helping students understand what this means in engineering terms. The livescript file *freq_responses_with_matlab.mlx* introduces users to the concept of frequency response and how the gain and phase might be computed from both time domain responses and with complex algebra (e.g. see Figure 1). The file provides coding for numerous examples to show a large variety of possible behaviours and also illustrations of how the gain and phase characteristics can be obtained from these figures (similar to Figure 2).

Nevertheless, as hinted earlier, a lecturer may begin with the app file *frequency_response_time_behaviour.mlapp* as this works in partnership with the livescript and allows users to explore and understand the concepts of frequency response in the time domain by providing an interactive interface, as shown in Figure 2. Here users can only select from a set of pre-entered systems (yellow box) and frequencies (green slider) and the figures and text boxes (bottom left) show the corresponding time domain based frequency response analysis. For completeness the app compares systems with and without a delay to help stu-

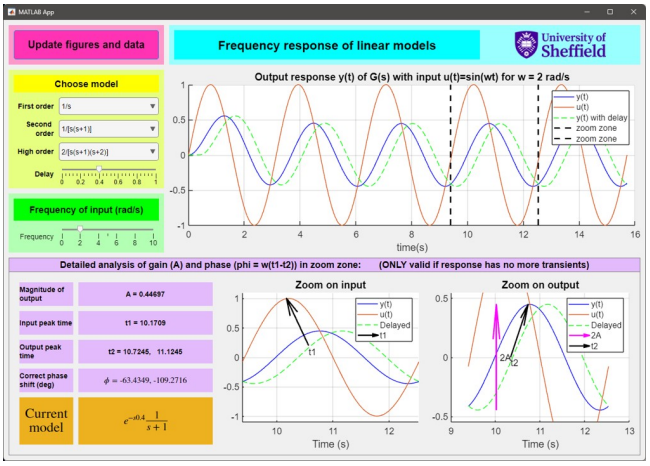

Fig. 2. Virtual laboratory interface on frequency response

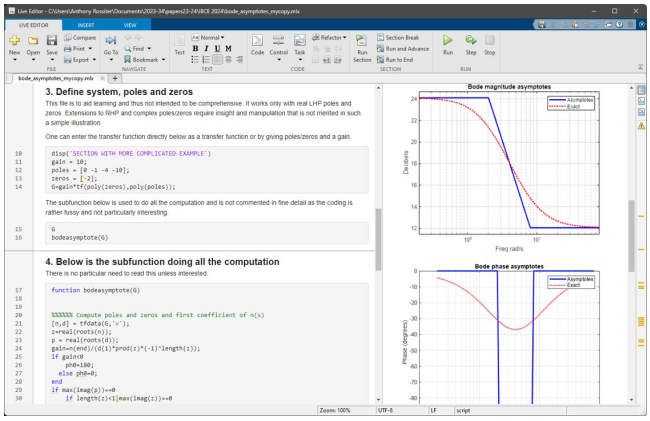

Fig. 3. Screen capture from file *bode_asymptotes.mlx*.

dents understand the repercussions of delay on frequency response, given this feature is common in many real processes. Once familiar with the concepts, students can use the livescript *freq_responses_with_matlab.mlx* to learn the required coding for individualised work.

3.3 Capturing frequency response data using Bode diagrams

While time domain plots are useful context, in practice these are not efficient mechanisms for deriving the frequency response over a wide frequency range and thus users are more likely to infer this from the available models. It is normal to use Bode diagrams as these are effective at capturing the large changes in frequency and gain (several orders of magnitude). Plotting Bode diagrams is somewhat tedious by hand, but it is useful for students to be able to do rapid sketches to improve their conceptual understanding of how poles and zeros affect the shapes and detail; this insight is used in the lead and lag compensation designs later in this paper.

Consequently the file *bode_asymptotes.mlx* focuses on supporting sketching which is typically done using asymptotic methods. It provides several illustrations (e.g. Figure 3), with supporting code, of the asymptote computations. Users can edit the code very easily to insert their own examples. As yet there is no virtual laboratory (.mlapp file) for this activity although one will be developed in the future as time permits.

Remark 3. For simplicity the phase asymptotes have simple step values of $\pm 90^o$ and it was decided that using more involved approximations with slopes a decade to each side of the corner frequency are not included for now due to the focus on simplicity.

3.4 Links between closed-loop behaviour and open-loop Bode diagrams and gain/phase margins and proportional design using phase margins

An assumption is made in the frequency response suite of files that the phase margin (PM) is the most important value to take from a Bode diagram as few, if any, designs make explicit use of the gain margin. Other important characteristics are the gain cross over frequency w_c (where the phase margin is computed) and the low frequency gain.

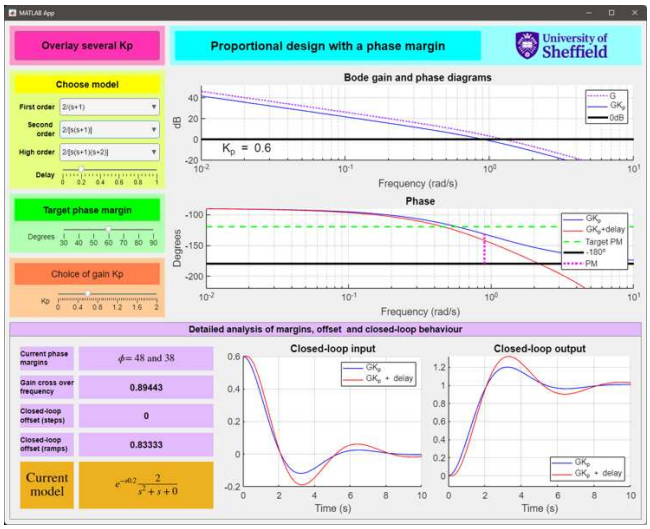

Fig. 4. Virtual laboratory interface on proportional design.

Given this, the resources are designed to emphasise these values and how they might change as a consequence of any design decisions.

The app file *proportional_design_phase_margin.mlapp* emphasizes the key values in the figures and purple text boxes (see Figures 4, 5). Viewers can see from Figure 5 how the PM changes (vertical pink dotted line) as the proportional gain is changed and also, in the bottom right axes, how this effects the closed-loop behaviour and cross over frequency w_c. In the main display of Figure 4, users can modify the proportional gain manually and thus find a value that delivers the desired PM (clearly marked with the green dashed line). Figures and data update automatically wherever anything is changed (e.g. system choice, gain choice, delay).

For users who want access to source code and examples, the livescript file *proportional_design_with_bode.mlx* provides this and through a combination of textual information, graphical illustrations and code, it guides users through the core design assumptions and steps.

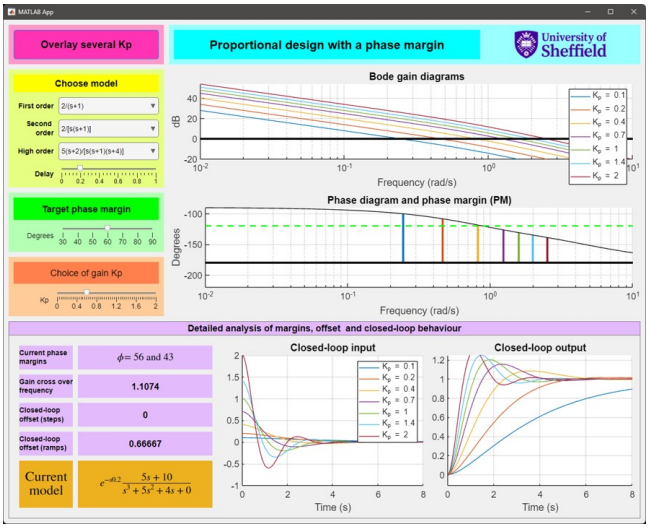

Fig. 5. Alternative virtual laboratory interface on proportional design.

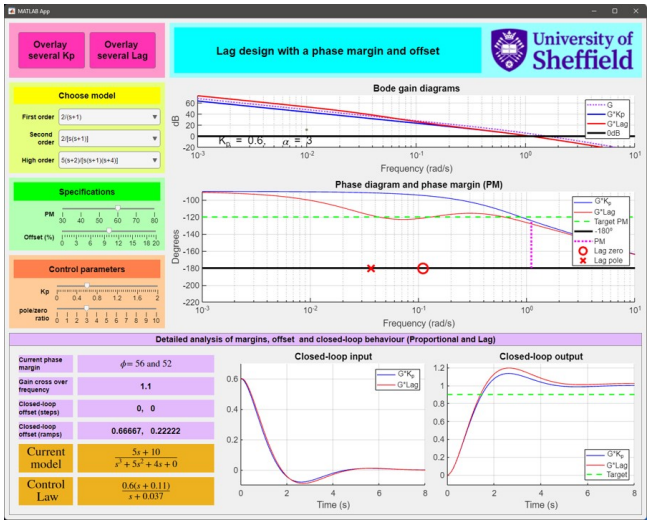

Fig. 6. Virtual laboratory interface on lag design.

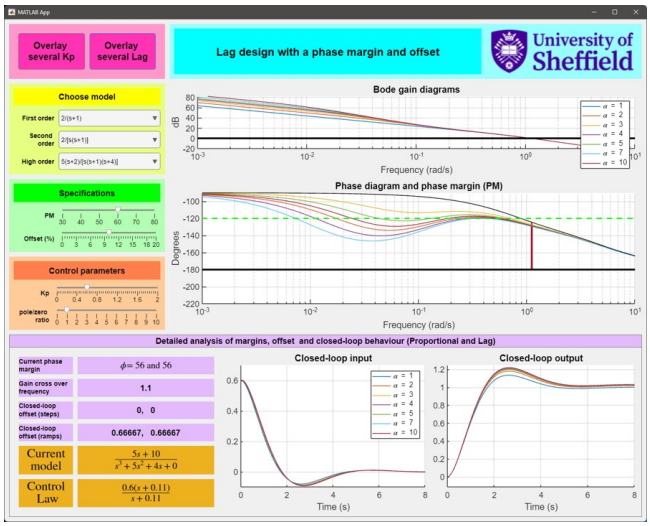

Fig. 7. Virtual laboratory interface on lag design showing different pole/zero ratios.

3.5 Lag compensator design using phase margins and low frequency gain

A lag design focuses on the low frequency gain and PM criteria (criteria defined in the green box, e.g. Figure 6). A lag compensator $M(s)$ (eqn. 1) has two degrees of freedom: compensator gain K_p and pole/zero ratio α (defined in the orange box).

$$M(s) = K_p \frac{s+z}{s+z/\alpha}; \quad \alpha > 1, \quad z = \frac{w_c}{10} \tag{1}$$

The poles and zeros are usually placed a decade below the cross-over frequency.

A simple mechanistic lag design follows on from proportional design in that the first step of a lag design is to undertake a proportional design. The second step is to add the lag component which increases the low frequency gain. Consequently, the virtual laboratory file *lag_design_phase_margin.mlapp* (Figure 6) includes similar functionality to the previous section (for example the 'overlap several Kp' button) and indeed setting the pole-zero ratio to unity gives just proportional control. In addition the 'overlay several lag' button allows the user to explore the impact of changing the pole/zero ratio (Figure 7). Consequently users can explore in an intuitive fashion the impact of changing the proportional gain and the impact of changing pole/zero ratio. The aim is to focus on achieving a target phase margin and low frequency gain (either for a ramp or constant offset criteria); core numerical numbers are displayed in the text boxes in the bottom left. Figures and data update automatically wherever anything is changed (e.g. system choice, gain choice, pole/zero ratio). The app automatically selects the pole and zero a decade below the gain cross over frequency (marked with bright red 'x' and 'o').

As in all the other examples, a livescript file *lag_design_with_bode.mlx* provides access to more detailed explanation of the core design steps, requirements definitions and the associated MATLAB code and several numerical examples. Students could use this file as a basis for a design on examples and requirements of their own choosing.

3.6 Lead compensator design using bandwidth and phase margins

A lead design focuses on the bandwidth and PM criteria (criteria defined in the green box, e.g. Figure 8). A lead compensator $M(s)$ (eqn. 2) has two degrees of freedom: compensator gain K_p and pole/zero ratio β (defined in the orange box).

$$M(s) = K_p \frac{s+w_c/\sqrt{\beta}}{s+w_c\sqrt{\beta}}; \quad \beta > 1 \tag{2}$$

The poles and zeros are chosen to bridge the desired cross-over frequency.

A mechanistic lead design can be used to achieve a specified gain cross over frequency (marked with a vertical green dashed line) and phase margin (marked with a horizontal dashed green line) and the virtual laboratory here is designed to help students understand the core steps in such a process, and implicitly to be more flexible should

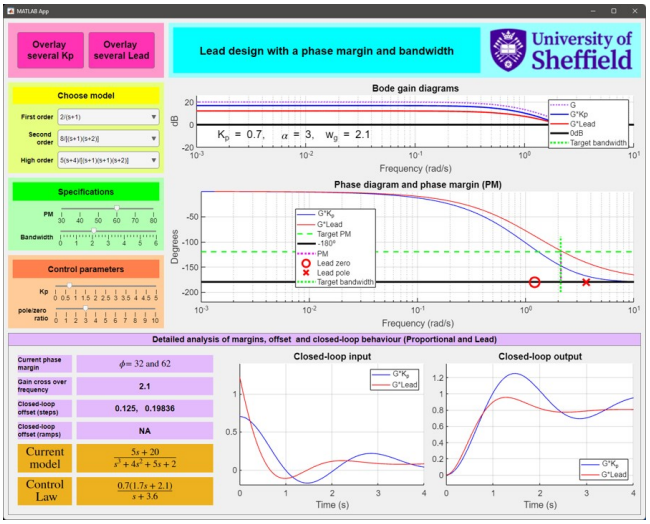

Fig. 8. Virtual laboratory interface on lead design.

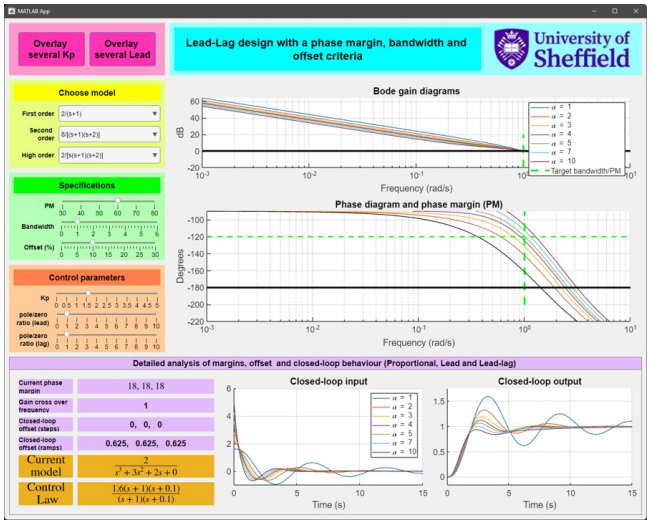

Fig. 9. Virtual laboratory interface on lead-lag design: overlaying mutiple pole/zero ratios.

they need to be. The file *lead_design_phase_margin.mlapp* allows a user to practise simple lead compensator design using frequency response methods and in particular to focus on achieving a target phase margin and bandwidth criteria.

First find a gain to meet the desired gain cross over frequency (marked with a vertical green dashed line) and then, explore the range of pole/zero ratios to see the impact on the resulting PM (equivalent to that shown in Figure 9). An example of the selected lead is given in Figure 8 where the design criteria are shown with the green lines and the selected pole and zero are in red (x and o). The actual cross-over frequency w_c is shown in pink and users will see this does not overlap with the green if an incorrect gain is chosen.

As with earlier examples, the livescript file *lead_design_with_bode.mlx* explains the core steps behind a mechanistic lead design (PM and bandwidth criteria) and illustrates with examples and gives the associated code so users can try on examples of their own.

3.7 Lead-lag compensator design

Both lead and lag designs have weaknesses in that they can tackle two performance criteria whereas often, a good design would consider three, that is phase margin/damping, offset and bandwidth (see green box in Figure 9). Consequently it is logical to consider a lead-lag design which has sufficient degrees of freedom to incorporate these three criteria. The degrees of freedom (given in the orange box) are compensator gain K_p, and pole/zero ratios α, β for the lead and lag components. The design steps match those in the earlier designs and thus the required virtual laboratory can be very similar. A lead-lag compensator structure is given in eqn. (3).

$$M(s) = K_p \left[\frac{s + w_c/\sqrt{\beta}}{s + w_c\sqrt{\beta}}\right]\left[\frac{s+z}{s+z/\alpha}\right]; \quad \beta > 1, \quad \alpha > 1 \tag{3}$$

The virtual laboratory allows a user to practise simple lead-lag compensator design using frequency response methods and in particular to focus on achieving, in the order specified: bandwidth, phase margin and low frequency gain, is in the file *lead_lag_design_phase_margin.mlapp*. The three design choices (gain and pole/zero ratios for the lead and lag components) are clear in the orange box of Figure 9. A logical process, as explained in the manual, is as follows:

(1) First do a lead design to achieve the desired bandwidth and phase margin. The steps exactly match those in the previous section and thus, in those respects, the app is quite similar; see Figure 9 for an exploration of different pole/zero ratios for the lead component with the bandwidth criteria enforced (vertical green dashed line).
(2) Once the lead component is known, select the pole/zero ratio of the lag component to achieve the desired low frequency gain characteristic (Figure 10). The poles and zeros of the compensator are clearly marked where one can see the lead values bridged around the bandwidth and the lag values a decade smaller.

The file *lead_lag_design_with_bode.mlx* explains the core steps and coding behind a mechanistic lead-lag design (PM, low freq. gain and bandwidth criteria) and illustrates with examples. This file also compares with a lag design and a lead design. Consequently, this file can be used as a code template for students wishing to carry out and evaluate their own designs.

Remark 4. PID compensation has not been included in the frequency response files for now, but the author is happy for others to help him fill this gap.

4. ADOPTION IN TEACHING AND EARLY EVALUATION/REFLECTIONS

The apps were only released very late in 2023 and thus it is too early yet to have a formal evaluation. However the author has used early variants of livescripts with two MSc cohorts (2022-23 and 2021-22) during an introduction to control module and it was clear during supported computer sessions that the students were using these actively and finding them useful. Indeed the author's

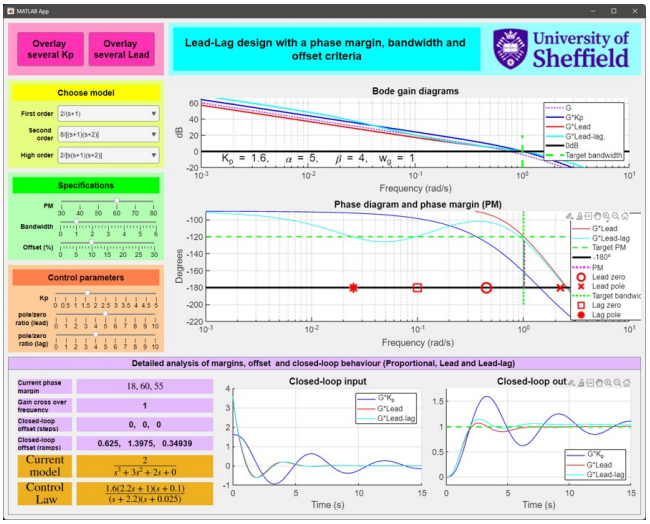

Fig. 10. Virtual laboratory interface on lead-lag design: final design.

impression is that their progress on undertaking lead and lag designs was significantly improved compared to previous cohorts who did not have these resources and relied mainly on *sisotool* and code snippets in m-file form.

More evaluation and integration into teaching will follow now that the apps have been developed and incorporated as the author expects these will be a core first step in both the lecturing and guided activities for future cohorts.

5. CONCLUSIONS

This paper has described some recently developed tools on an introduction to frequency response design, as would be appropriate for the end of a 1st course in control, or the beginning of a second course. The focus is on helping students visualise and understand the core design steps for lead, lag and lead-lag compensation so they can appreciate the logic in how both gain and pole/zero ratios are selected to meet well defined performance criteria. The tools can easily be used for demonstrations during lectures, or for students to learn through trial, error and guided study. They have now been integrated into the control101 toolbox and thus are available through the apps button once the toolbox has been downloaded.

Nevertheless, it is clear that different institutions and degrees may want to cover other aspects, illustrations and design methods so the author would like to encourage those persons to contact him to work together to both improve the existing resources and also to create more resources that can be added. He is already aware of a few colleagues who have interest and have worked with him and we hope their contributions will be included in any future toolbox release.

REFERENCES

Cameron, I., 2009, Pedagogy and immersive environments in the curriculum, Blended Learning conference, 290-294.

de la Torre, L., R. Heradio, C. A. Jara, J. Sanchez, S. Dormido, F. Torres, and F. Candelas, 2013, Providing Collaborative Support to Virtual and Remote Laboratories. IEEE Transactions on Learning Technologies.

de la Torre, L., Neustock, L.T., Herring, G.K., Chacon, J., Clemente, F.J.G., and Hesselink, L., 2020, IEEE Transactions on Industrial Informatics, Automatic Generation and Easy Deployment of Digitized Laboratories, 12, 16, 7328-7337, doi = 10.1109/TII.2020.2977113

Dorf, R. and Bishop, R., 2021, Modern Control Systems, Pearson Education, Limited, ISBN: 9781292422350

Douglas, B., 2022, Resourcium, https://resourcium.org/

Golnaraghi, F. and Kuo, B.C., 2017, Automatic Control Systems,Mcgraw Hill,ISBN: 978-1259643835

Guzman, J.L., Piguet, Y., Dormido, S., Berenguel, M., Costa-Castello, R., 2018, New Interactive Books for Control Education, IFAC papers online, Volume 51, Issue 4, 2018, Pages 190-195.

Guzman, J.L., Berenguel, M., Dormido, S., Costa-Castelló, R., 2023, Using interactive tools to connect theory and practice, IFAC-PapersOnLine, vol.56, no.2, pp.9606.

Matisak, J., Pohancenik, M. and Zakova, K., 2023, Platform for Virtual Laboratory of Mechatronic Systems in Augmented Reality, EXPAT 2023 and IEEE Explore

Nise, N.S., 2017, Nise's control systems engineering, Wiley, ISBN: 9781119382973

Ogata, K., 2009, Modern control engineering, Prentice Hall, ISBN: 9780136156734

Rossiter, J.A., 2017, Using interactive tools to create an enthusiasm for control in aerospace and chemical engineers, IFAC world congress (ifacpapersonline).

Rossiter, J.A., B. Pasik-Duncan, S. Dormido, L. Vlacic, B. Jones, and R. Murray, 2018, Good Practice in Control Education, European Journal of Engineering Education, http://dx.doi.org/10.1080/03043797.2018.1428530 .

Rossiter, J.A., Serbezov, A., Visioli, A., Zakova, K. and Huba, M., A survey of international views on a first course in systems and control for engineering undergraduates, IFAC Journal of Systems and Control, Vol. 13, Article 100092, 15 pages, 2020. https://doi.org/10.1016/j.ifacsc.2020.100092

Rossiter, J.A., 2021, Modelling, dynamics and control website, http://controleducation.group.shef.ac.uk/mainindex.html

Rossiter, J.A., 2022, MATLAB apps to support the learning and understanding of simple system dynamics, IFAC Symposium on Advances in Control Education (ACE 2022).

Rossiter, J.A., Control 101 toolbox, 2023, https://controleducation.sites.sheffield.ac.uk/matlabresources/section-6-11-community-control-toolbox or https://github.com/jarossiter/control101

Rossiter, J.A., A suite of MATLAB livescript files to support learning of elementary control and feedback concepts, IFAC world congress, 2023.

Serbezov, A., Zakova, K., Visioli, A., Rossiter, J.A., Douglas, B. and Hedengren, J., 2022, Open access resources to support the first course in feedback, dynamics and control, IFAC Symposium on Advances in Control Education.