

A MATLAB GUI for learning controller design in the frequency domain

Conference or Workshop Item

Accepted Version

Mitchell, R. (2014) A MATLAB GUI for learning controller design in the frequency domain. In: 2014 UKACC 10th International Conference on Control, 8th - 11th July 2014, Loughborough University. Available at <http://centaur.reading.ac.uk/36801/>

It is advisable to refer to the publisher's version if you intend to cite from the work. See [Guidance on citing](#).

Published version at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6915153&tag=1>

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the [End User Agreement](#).

www.reading.ac.uk/centaur

CentAUR

Central Archive at the University of Reading

Reading's research outputs online

A MATLAB GUI FOR LEARNING CONTROLLER DESIGN IN THE FREQUENCY DOMAIN

R.J.Mitchell

School of Systems Engineering
University of Reading
Reading, UK
r.j.mitchell@reading.ac.uk

Abstract—A MATLAB GUI is presented which is used to help students learn to design controllers in the frequency domain. It complements the author’s two previous GUIs for plotting and identification of systems in the frequency domain. It also incorporates the concept used in the “electronic calculator that makes students think” to assist learning. Positive student feedback affirms that the GUI has helped their understanding.

Keywords- Education, Frequency Response, Bode diagrams

I. INTRODUCTION

Frequency Response methods are an integral part of many engineering degrees, typically taught as part of the second course in control. Programs such as MATLAB can do many of the associated calculations, but students may not fully appreciate what is being done. This is explained in [1]:

“I think there is great value in teaching the theory of Bode diagrams and Nyquist plots. This provides the basis of understanding the plant dynamics and the effects of closed loop control and the controller settings. However, although I think students should understand the mechanics of calculating the frequency response of a system, the use of MATLAB should enable the calculations to be done quickly so that time can be devoted to controller design. It would also allow higher order systems to be investigated.”

To help the understanding of dynamics and frequency response, the author has developed two GUIs for a set of ten third year lectures on Frequency Response, given to students studying Cybernetics, Robotics and Electronic Engineering, [2], [3]. In one GUI the student interprets the transfer function of a system by specifying the asymptotes of the Bode plot. As teaching identification also helps students to understand frequency response [4], a second GUI is used, where the student identifies the system from Bode plots.

Both GUIs operate in two modes, a simpler mode for familiarization, and then a more advanced mode. The two modes help reinforce learning, which student feedback confirms. This is consistent with Hayden [5], who notes that it is beneficial to present material in complementary approaches.

The GUIs form the basis of about 60% of the coursework for the module, helping students to understand frequency responses. The remaining 40% is for controller design.

In the past, the author has required students to write code to implement the control algorithms. This has some disadvantages including the potential for plagiarism. But more fundamentally, the focus was on the algorithm rather than on doing the designs. As Atherton notes, design is “an iterative process and students should be allowed to use modern software to do this” [6]. This should be the focus of the work.

As such, and given the success of the other GUIs, the author has developed a GUI aimed at helping the students learn about the design of controllers in the frequency domain. This GUI, like its predecessors, operates in a simpler and then a more advanced mode. The former is used for familiarization and some simpler design tasks are set. In the advanced mode, students have to use the GUI iteratively to design controllers so the system can achieve specific specifications. In addition, rather than relying on MATLAB to do the calculations, a technique employed in the “Electronic Calculator that makes students think” is used [7] whereby students have to make good enough estimates of key values in order to do the design.

This paper is arranged as follows. First are described some of the key aspects of the assignment in previous years and the author’s reflections influenced by student feedback. Then is described the new GUI for controller design and how it forms part of the assessment. Positive feedback and supportive comments are given and some conclusions drawn.

II. THE ASSIGNMENT

The learning outcomes for the course are an understanding of the frequency response of linear systems, the ability to plot these, to identify systems from the response, and to design controllers in the frequency domain to meet a specification.

The module coursework has three parts: plotting asymptotes and identifying systems, using the GUIs, and designing controllers, where students wrote code. Word documents are provided for each part, containing a table which specified the various tasks to be performed with a built in marking scheme. Students paste the output from MATLAB, such as graphs, any code they had written, or output from the GUI, into the relevant parts of the table, and also answer specific questions to help them reflect on their answers and what they had learnt, and provide feedback to the author.

The students are provided with the MATLAB files associated with the GUIs, and an extra m file used throughout the assignment. The main task of the extra m file is to return the system whose system number is specified in the GUIs.

In the past, many of these systems were the same for all students, but in an attempt to avoid plagiarism, the gain and corner frequencies of two systems were determined from the student's unique eight digit student number.

The m file also defined functions used to design suitable controllers. That for a Proportional controller was provided, to demonstrate key aspects of the code. Students were required to complete those for Phase-Lead, P+I and PID controllers.

Although the GUIs worked well, receiving positive feedback from the students, and the marks were good, the students did poorer than expected in the examination. This may have been because students were working together on the assignment. On reflection, it was a mistake that, for much of the assignment, all students plotted or identified the same system. As students had to write only a few lines to implement each controller, extending that provided for the proportional controller, it was inevitable that code looked similar. Another issue was that students on some degrees believed they had done less MATLAB programming than others, and so felt disadvantaged. A better approach was needed.

A fundamental issue, however, was that the focus was too much on coding the algorithm and not enough about understanding and using the methods for different designs.

The author thus determined to make three changes. First, all systems used in the assignment would be unique to each student. Second, a GUI would be provided for designing controllers, based on the author's experience of the plotting and identification GUIs. Third, more emphasis would be on performing actual designs, iteratively.

The GUIs require students to enter their eight digit student number and a system number. The latter defines the structure of a system, but the overall gains and corner frequencies are calculated from the student number.

For the controller GUI, the author wanted a system where the student had to understand the approach. As such he noted the article "Educational Calculator Makes Students Think" reported by Berger (2013), which states that "a calculator yields answers but not necessarily understanding". To encourage mathematical understanding a student not only enters a calculation to be performed, but also has to estimate what the answer should be. Only if the answer is close enough (and the tolerance is wider the more complex the calculation), is the true answer is displayed.

For the controller GUI, key values in the process are determined from the Bode plots. MATLAB can easily produce these, but that does not necessarily help understanding. In the GUI, students are asked to estimate values typically from the Bode plot, and these are accepted only if they are close enough.

The next section briefly outlines the four design methods used. This is followed by a description of the GUI.

III. CONTROLLER METHODS

The GUI supports the design of four controller types: Proportional, Phase Lead, P+I and PID, though more could be added easily. All four methods are to be designed in the frequency domain, as described below, where $P(j\omega)$ is the process to be controlled and $C(j\omega)$ is the controller.

For the proportional controller, $C = K_p$, and the designer specifies the desired phase margin, PM. The angular frequency where phase of P is $-180^\circ + PM$ is determined: call this ω_x . The gain of $P(j\omega_x)$ is then found. K_p is the inverse of this gain.

The phase lead controller is of the form:

$$C(j\omega) = K_p \frac{1 + j\omega T_e}{1 + j\omega T_a} \quad (1)$$

The designer again specifies PM and also n the factor by which the system is to be sped up. Again ω_x is determined where the phase of P is $-180^\circ + PM$. The phase of $P(jn*\omega_x)$ is then found: call this P_n . At $n*\omega_x$ the controller must provide phase lead equal to the difference between P_n and $-180^\circ + PM$. The time constants are then determined using the following

$$\alpha = -180^\circ + PM - P_n$$

$$G = \tan\left(45^\circ - \frac{\alpha}{2}\right) \quad (2)$$

$$T_e = \frac{1}{G * n * \omega_x}$$

$$T_a = \frac{G}{n * \omega_x}$$

The gain of the controller, K_p , is then found using the proportional control method operating on P combined with the lead and lag terms.

The P+I controller is of the form

$$C(j\omega) = K_p \frac{1 + j\omega T_i}{j\omega T_i} \quad (3)$$

The designer specifies PM together with a working angular frequency ω_c which is typically where the phase of P is around -90° . The controller is calculated such that the gain of $C*P$ is unity and their phase is $-180^\circ + PM$ at ω_c . Defining

$$\phi = -\pi + PM - \angle P(j\omega_c) \quad (4)$$

But,

$$\phi = \angle C(j\omega_c) = \tan^{-1} \omega_c T_i - 90^\circ \quad (5)$$

Thus,

$$T_i = \frac{1}{\omega_c} \tan(\phi + 90^\circ) \quad (6)$$

Again K_p is found by applying the proportional control method to P combined with that part of C which has already been determined.

The PID controller is designed using the modified Zeigler-Nichols method, where the derivative time constant is a quarter of the integral constant. As such the controller is of the form:

$$C(j\omega) = K_p \frac{(1 + j0.5\omega T_i)^2}{j\omega T_i} \quad (7)$$

The designer specifies PM together with a working angular frequency ω_c which is typically where the phase of P is around -180° . The controller is calculated such that the gain of C*P is unity and their phase is $-180^\circ + PM$ at ω_c . Here ϕ is determined as in (4) and T_i is computed by

$$T_i = \frac{2}{\omega_c} \frac{\cos(\phi)}{1 - \sin(\phi)} \quad (8)$$

T_d is then $T_i/4$ and K_p is determined as before.

IV. THE CONTROLLER GUI

The GUI has boxes for the user to enter their student number, the system to be selected and the specification (desired Phase Margin and speed up factor if appropriate). The user selects a design method from those in a list box. The GUI also has various buttons for initiating designs, copying to the clipboard, etc. When a system is selected, its transfer function is given and its Bode plots are displayed. Later, when a design is performed, the closed loop step response is also shown.

When the user presses the DoDesign button, a panel appears in which the user is asked to estimate three items in turn. For each, the user is asked to estimate a particular value. If it is close enough, the correct value is automatically inserted and the next question asked. ‘Close enough’ is determined according to the ease with which the estimate can be found, as is illustrated in the examples given below. If a poor estimate is given, the user must try again. The help button can be pressed, if needed, and a suitable hint is displayed. When the three questions are answered correctly, the associated controller is displayed and the user can accept it or cancel the design.

For the proportional controller, the user has first to enter the angular frequency where the phase is $-180^\circ + PM$: the help hint for this suggests the user looks carefully at the phase plot. The tolerance on the value is around a factor of 1.5. Next the user is asked to estimate the gain at this angular frequency. A factor of 5 is allowed for this as the gain can operate over a wide range. Finally, the user is asked to estimate the controller gain (which is the reciprocal of the previous answer) As this is a simple calculation, a smaller tolerance is allowed.

Fig 1 shows the GUI after the user has estimated that the angular frequency is 50, which is deemed close enough to the actual value of 51.68.

Once the three questions have been answered and the controller is accepted, the controller parameters are installed in the space for C(s), and the step response is plotted, in which the peak value, time to peak and settling times are shown.

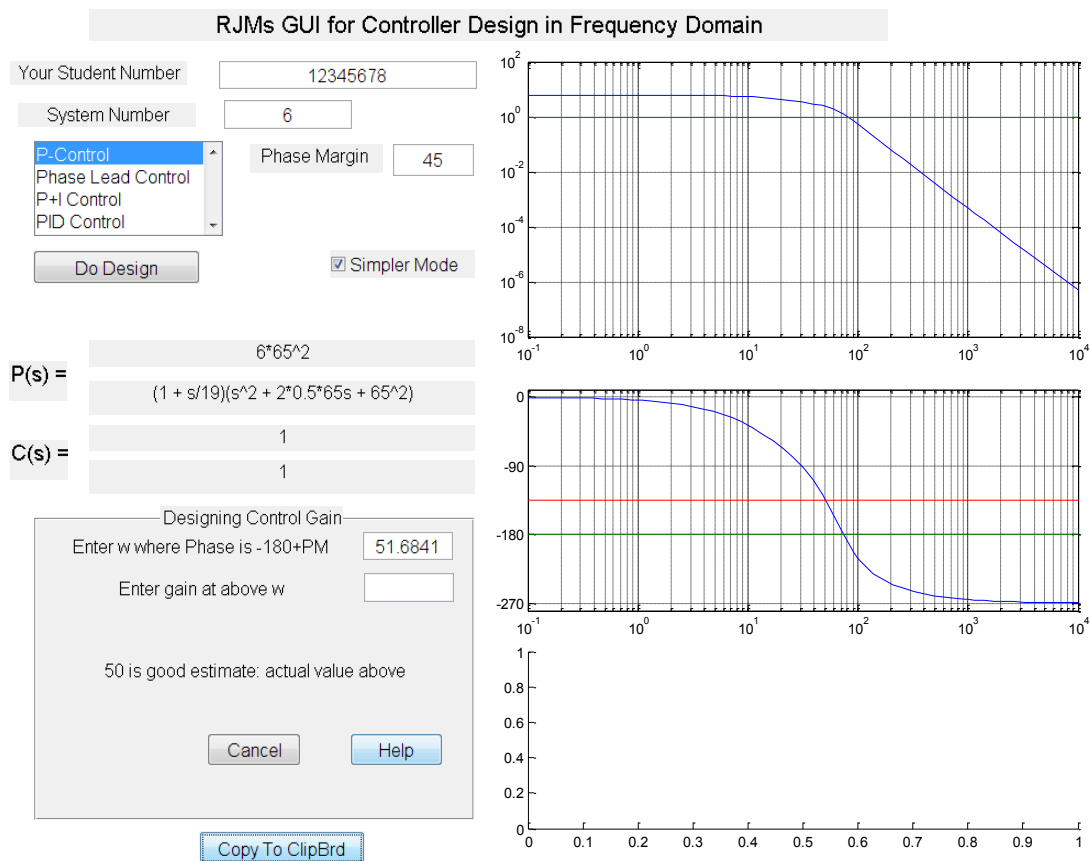


Figure 1 Controller GUI after the first question has been successfully answered

Fig 2 shows the parts of the GUI changed at this point, namely the controller is shown and the step response plotted.

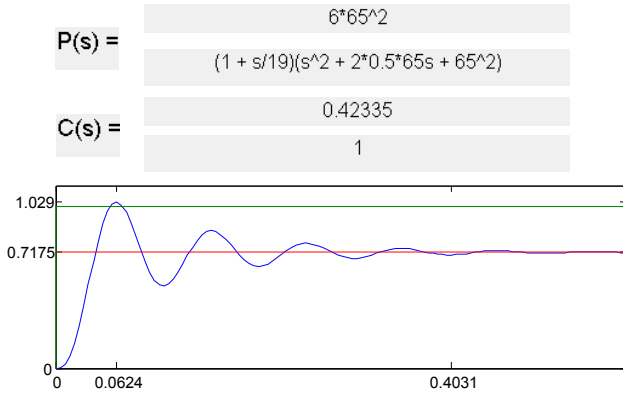


Figure 2 Updated Parts of the GUI

If the Phase Lead Control option is selected, the user is able to specify by how much the response should be speeded up. The student estimates the angular frequency where the phase is $-180^\circ + PM$, the phase at this frequency times the speed up factor and the amount of phase lead the controller must produce. When these are entered, the system calculates the value of the time constants and the user can decide to accept them or cancel the design.

The student is not asked in the GUI to calculate the actual time constants, as the aim is to ensure the student understands the process associated with doing the design. However, in the

assignment, when students are designing a Phase Lead controller for one system, in addition to pasting the output from the GUI into the word log, they are also asked to show the calculations necessary to calculate T_e and T_a and to compare their results with those in the GUI.

Fig 3 shows the panel in the GUI at this stage, for the same system as depicted in Fig 1, with 45° phase margin and a speed up factor of 2.

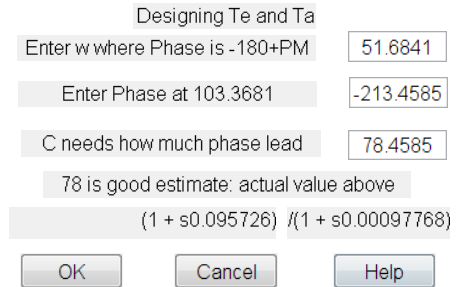


Figure 3 Panel for determining time constants

Once these time constants are accepted, the controller panel is updated to reflect the time constants, the Bode plots are updated, the step response is shown, and then the GUI enters the mode whereby a proportional design is performed. Fig 4 shows the GUI at the stage when K_p has been calculated: note the step response is that for which T_e and T_a have been incorporated, but K_p has not been determined.

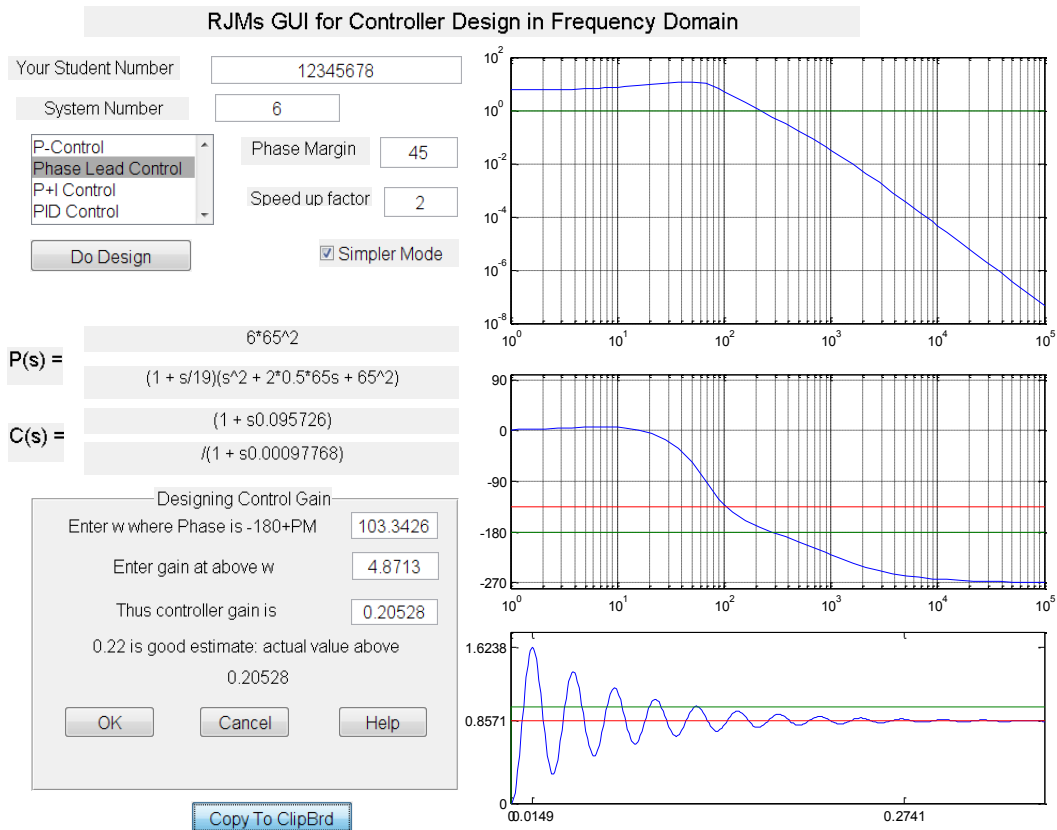


Figure 4 GUI when determining K_p for Phase Lead controller

For P+I control, the user is asked first to enter the phase, p , associated with the working angular frequency ω_c . This is typically -90° , but the user can specify other values. The user then estimates ω_c by inspecting the phase plot. The third question asks for ϕ , the required phase of the controller, which should be $-180^\circ + PM - p$. The GUI then calculates T_i and the user can accept or cancel it. Once accepted, the controller and Bode plots are updated, the step response is shown, and the GUI enters the mode for determining K_p . For one system, in the assignment, the user is asked to record in their log the calculations needed to determine T_i from ω_c and ϕ .

For PID control the same process is used, except the initial phase is typically -180° , and once the value for ϕ has been entered, the values for T_i and T_d are calculated and displayed for acceptance by the user. The GUI then incorporates these values, updates the Bode plot, shows the step response and enters the mode for calculating K_p .

V. ADVANCED MODE

In the simple mode of operation, if a PID controller is to be designed, say, the time constants are found and automatically the GUI sets in motion the calculation of the gain. In the (slightly) more advanced mode the student determines the order in which the steps are taken. The listbox now lists the actions and there are two design buttons: "Start Design" which causes $C(s)$ to be reset to unity before the design is performed and "Add to Design" where the design builds on $C(s)$. Fig 5 shows these. Then, for instance, to do a PID design, students are expected to select "Find T_i & T_d for PID", press Start Design, then select "Find Gain" and press Add to Design.

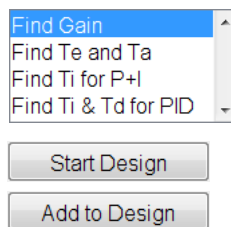


Figure 5 Listbox and Buttons in advanced mode

Although this is only slightly more advanced, the assignment tasks have been selected to be more difficult when this mode is used, expecting the students to iteratively use the GUI until the design meets a specification.

VI. DESIGN ASSIGNMENT

The controller GUI is used in the third part of the assignment for the module, and is worth 40%. The tasks set where the GUI operates in simple mode are given below. Note system 2 is a second order type 0 system; system 4 is type 1, with one zero and two poles; and system 5 is type 0 with 3 poles. The actual gain and corner frequencies are determined from the student number. The tasks are as follows

P-controller for system 2 with $PM = 45^\circ$

Phase Lead controller, system 2 : $PM = 45^\circ$, speed factor 2

Students are asked to show how T_c and T_a are calculated.

P+I controller system 2 : $PM = 45^\circ$

Students are asked to show how T_i is calculated

P-controller for a system 4 with $PM = 60^\circ$

Phase Lead controller, system 4 : $PM = 60^\circ$, Speed factor 3

P+I controller system 5 : $PM = 60^\circ$

PID controller system 5 : $PM = 60^\circ$.

Students are asked to comment on their results.

In advanced mode, the students work on system 13, which is type 1 with two poles, and system 15 which is type 0 with one zero and four poles. The tasks are to design the following:

P-controller on system 13, with overshoot $20\% \pm 0.5\%$

Phase lead-controller, system 13, same overshoot, whose peak time is 40% less than that with the P-controller.

P-Controller on system 15, with $PM = 45^\circ$

System 15, controlled to have 20% overshoot, zero steady state error, at least as fast as the previous design.

For the advanced mode, the students are expected to perform suitable designs, with different values of PM, say, in order to meet the specifications. Precise specifications of peak time, for instance, are not possible as the corner frequencies of the systems are different for all students.

VII. EVALUATION AND REFLECTION

In addition to pasting values and graphs into a structured report, students answer questions about the assignment and its GUI. Their answers and notable comments are given below.

What do you think of the GUI?

All students were positive about the GUI, with some useful suggestions for improvement.

Do you like the fact that you are asked to estimate values which are accepted/corrected if they are close enough and rejected otherwise? Does it help your understanding? Does the GUI accept values which are too far away?

With one exception, all students liked this feature.

"Yes it helps in the obvious way but I also believe that the constant feedback builds confidence in your methods and if you are wrong then it allows you to rethink rather than panic that what you have done is incorrect."

"I think this is quite a good feature of the GUI, and it does make the user think about what is actually required in the designing of control systems, although it does occasionally accept values that seem a little bit too far out of the range."

"I feel that this is a really useful aspect of the GUI as it provides a forgiving but still instructive introduction to the topic. The student has to have a basic understanding of what is going on before they can find the correct value. The hints also add an initial level of explanation that can really help the initial uses of the GUI."

The GUI calculates the time constants for you – is that right or should you be asked to calculate them, either exactly or estimating values?

All who expressed a view were supportive, for the simpler mode, at least. One student commenting:

“For speed, having the time constants calculated is helpful. For extra understanding it would be best to ask students to calculate them manually- perhaps as part of the ‘non-simple’ mode. The estimation process is very useful...”

Was it useful to you to have been asked to calculate T_e/T_a and T_i in the design?

All agreed that this was a good idea. One student saying:

“Yes very, made me think critically about what actually needs to be done and more importantly why ...”

From the point of view of your learning is it better to have the two modes of design? Justify your answer.

All said yes. Interesting justifications included:

“I believe that it is beneficial ...as it allows the system to be investigated at different levels. At the simpler mode the behaviour of the system can be seen and at the advanced mode the skills obtained in the simple mode can be translated into understanding for when the system is designed. This creates an element of progression....”

“The GUI also forces you to re-input values as you iteratively improve the system. By doing this the process of calculating the necessary values ... is reinforced.”

“Yes, as it gives the ability to draw a comparison between the two and thus understand better the difference, advantages and disadvantages of both.”

Has the Design GUI helped your understanding? Justify.

All students answered yes. Notable comments were:

“Yes, as it has allowed to “play” with the design of systems and obtain a better feeling about what parameter change is having what effect on the final system.”

“The GUI has certainly helped my understanding of control systems design. It serves as a good graphical representation of the effects that different control systems have, being able to add different types of control & seeing the effect certainly helped my understanding.”

In the past, students were asked to write short MatLab functions to do the different designs. Do you feel that the GUI is a better way of learning? Would you prefer to write code?

Most students preferred the GUI, though some would have liked the opportunity to program the controllers. The following is particularly pertinent.

“Very little time on the course has been spent writing MATLAB code. Given how much has been written for us, ... it would be very useful to practice writing some ...”

“However, as the purpose of this course is to learn about frequency responses, the GUI is better ...”

VIII. REFLECTION

As hoped, students believe the GUI has helped their understanding of controller design, and the impact the controllers have on systems.

The student feedback will inform improvements to the GUI and the assignment. Students will be allowed to write and submit code to calculate key time constants. In advanced mode, students may be asked to enter estimates of the time constants.

It was interesting that one student, for the final design, implemented a P+I followed by a Phase-Lead controller, rather than a PID controller. This may be followed up.

The author is developing web pages for resources such as these GUIs. These can currently be found on a link from <http://www.personal.reading.ac.uk/~shsmchlr/teach.htm> which could be linked with the shared resource web site given in [1].

The author is encouraged by the following student quote:

“Although not felt in a tangible way there is a considerable psychological effect on a student using a GUI that has been specially written by a lecturer to promote their understanding of the subject. Having a tool like that promotes goodwill between the lecturer and the pupil as they can see that the lecturer is willing to put in a significant amount of effort creating this tool for them and may therefore be seen as more approachable”.

IX. CONCLUSION

A third GUI has been developed aim at helping students to think and learn about the design of controllers in the frequency domain. It complements and builds on the authors previous GUIs for plotting and identification. Students believe that the GUI has helped them to learn about designing controllers.

REFERENCES

- [1] J.A. Rossiter, D. Giaouris, R. Mitchell., and P. Mckenna, ‘Typical control curricula and using software for teaching/assessment: a UK perspective’. Proc 17th IFAC World Congress, Seoul, Korea, 2008, pp 10331-6
- [2] R.J. Mitchell. ‘Using MATLAB GUIs to improve the learning of frequency response methods’, Proceedings of UKACC Control 2012 Cardiff, UK. 2012 pp 1031-1036.
- [3] R.J. Mitchell. ‘Improved MATLAB GUIs for Learning Frequency Response Methods’, Proc. 10th IFAC Symposium Advances in Control Education The International Federation of Automatic Control, August 28 - 30 2013, University of Sheffield, Sheffield, UK, pp. 303-308.
- [4] Robert D. Throne. ‘Frequency Domain System Identification of One, Two, and Three Degree of Freedom Systems in an Introductory Controls Class’, Proc. American Society for Engineering Education Annual Conference & Exposition, Paper 2005-493, 2005.
- [5] B. Hayden, ‘Teaching to variation in learning’, from http://www.brown.edu/Administration/Sheridan_Center/publications/handbooks/vari_learn.pdf (2006)
- [6] D.P. Atherton, (2006) ‘Some thoughts on the first course in control engineering’. In: 7th IFAC Symposium on Advances in Control Education (2006). Advances in Control Education, 7 (1). Curran Associates Inc. ISBN 9783902661074,
- [7] I. Berger, ‘Educational Calculator Makes Students Think’ <http://theinstitute.ieee.org/technology-focus/technology-topic/educational-calculator-makes-students-think>, 2013.