Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 53-2 (2020) 17604–17609

# Using interactive tools to facilitate student self-testing of dynamics and PI compensation

**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:** Virtual laboratories have become a common tool in recent years for supporting student learning and engagement. This paper presents a new tool for helping students self-assess their competence in basic dynamics for 1st and 2nd order systems alongside simple PI compensation techniques. The tools provide a supported environment for helping students work towards the correct answer by providing succinct feedback on incorrect responses and opportunities to try again, while displaying relevant information. A partner interactive tool is also provided which focuses solely on assessment with no feedback, so that students can assess their ability to get correct answers in a scenario that only the first attempt counts. This paper gives the thinking behind the tools, their coding and also accessibility for students.

*Keywords:* Virtual laboratories, staff efficiency, student engagement, independent learning.

## 1. INTRODUCTION

The production and use of remote and virtual laboratories has been growing in recent years (Dormido et al, 2012; Cameron, 2009; Abdulwahed, 2010) to the extent that the concept can now be considered mainstream within the global community. Part of this is driven by the realisation that hardware laboratories are expensive and thus access to these by students will always be limited (DIAMOND, 2016; STEMLAB, 2016) and thus, the only way to increase access to authentic laboratory experiences and trial and error learning, is to make some access available online or computer based. Ideally staff would like students to spend more time with hardware so that:

(1) Students can experiment with parameters and simulations without the time pressure of a timetabled laboratory.
(2) Students are enabled to be independent learners and ask *what if* questions and perform tests outside of tightly defined learning outcomes, for example to support assignments.

### *1.1 Remote and take home laboratories*

Some authors have focussed primarily on remote access to hardware laboratories (de la Torre et al., 2013; Qiao et al., 2012) and where this can be done, it is clearly authentic and valuable. However remote laboratories (RL) have a number of obstacles such as:

- RL are expensive to design and build. Inevitably they require significant ongoing support from technical staff, continual software updates and intensive monitoring during periods of heavy usage (Chen et al., 2010; Vargas et al., 2011).
- RL have relatively poor accessibility as they are single user by definition, especially where the associated equipment dynamics are slow. An effective online queueing system is needed, but even with this they may be inviable for large classes who all want access in the same few days (Rossiter et al., 2011).
- Most departments do not have the expertise or resource to support RL and an enthusiast is usually needed to get things going (Fabregas et al., 2011; Rossiter et al., 2018).

Another alternative that is growing in popularity is the concept of take-home-labs (Beal et al., 2018; Hedengren, 2019; Stark et al., 2013; Rossiter et al., 2019). The main idea here is that students can *borrow* their own personalised piece of equipment for several weeks and thus experiment as they wish. A good take home laboratory needs certain attributes (Rossiter et al., 2018):

(1) The hardware should be plug and play in a standard laptop and software freely available to the student.
(2) Staff provide code templates which are easy to edit by students to create their own tests and parameter choices.
(3) Support resources should be sufficiently high quality to enable independence.

Nevertheless, the efficacy of both remote and take home laboratories does not mean that the majority of academics are creating and utilising such activities and indeed the number of persons active in this area is still quite small, as evidenced by the size of conferences such as the Symposium on Advances in Control Education. The reasons are quite simple: most academics are recruited primarily to do technical research and thus do not have the capacity to develop and maintain the requisite expertise to design and develop high quality hardware laboratories. Indeed, the

2405-8963 © 2020, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2020.12.2677

most successful examples are typically developed by teams which include members whose prime role is education (Fabregas et al., 2011; Perez et al., 2011).

### *1.2 Virtual laboratories*

This scenario leaves somewhat of a conundrum for many conscientious academics who see the huge potential of interactive resources to faciltate student self learning and assessment, but have limited time themselves to lead the development. An obvious alternative are so-called virtual laboratories (VL). These are far more accessible in principle (assuming software availability), are not affected by slow run times and, in priniciple, allow parallel access by very large numbers. Moroever, they are robust as they are not subject to hardware faults and failures. Consequently a large part of the community has developed many VL. e.g. (Cameron, 2009; Goodwin, 2010; de la Torre et al., 2013; Fabregas et al., 2011; Perez et al., 2011; Guzman et al., 2006; Rossiter, 2012).

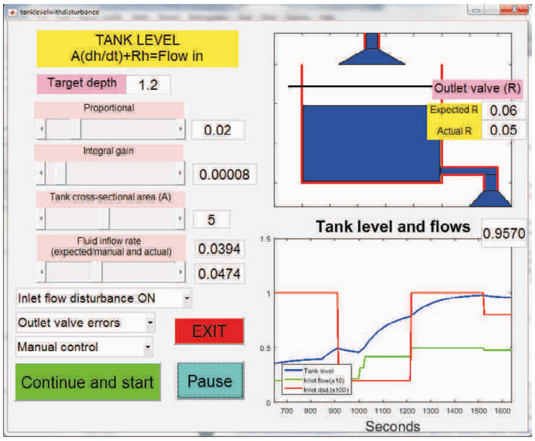
This paper is not going to focus on the high quality end of the VL provision, which inevitably requires time, good software and expertise, and rather makes the assumption that something that is quick and simple to create is better than nothing at all. Moreover, it is usually the case that the curriculum and learning outcomes vary from institution to institution so that often a resource is needed that can be easily tailored to the local requirements. High quality resources may come with specific software requirements, cost implications (Goodwin et al., 2011), accessibility issues due to the local environment and also, are not usually possible to edit locally without substantial expertise. Hence, as with earlier work, the assumption taken here is that VL will be developed (Rossiter, 2016) using MATLAB because:

(1) In the author's institution, all students to have access to MATLAB both on the network and for personal computers.
(2) Most control academics are familiar with MATLAB so will be easily able to edit any code for their own purposes.
(3) The GUIDE environment is reasonably intuitive and straightforward to use, thus uses expertise most have already.

Hence, the argument is that students will accept the relative crude drawings/animations as these will still be sufficient for them to relate to the real scenario (Rossiter, 2017). The embedded capabilities of MATLAB means the author can focus on core learning outcomes and need not spend time on how to produce plots in a web browser, or do complex computations in C code or java and so forth. The author's estimate is one can do a basic GUI for a given learning outcome in about half a day or slightly longer where the learning outomes are more involved. Editting a provided GUI to local needs may only take an hour or less. These timescales are achievable for most staff to engage.

### *1.3 Contribution summary*

In summary, the main contribution of this paper is the presentation of a new VL designed to promote student self-assessment (Munoz de la Pena et al., 2012; Farias et al., 2016; Olivier et al., 2001) and learning of 1st and 2nd order dynamics and simple PI design. Students are able to enter a number of numerical answers to typical questions on these themes and will receive feedback in numerous forms that can be used to either validate their progress or encourage them to try again with a suitable edit.


tanklevelwithdisturbance

TANK LEVEL
A(dh/dt)+Rh=Flow in

Target depth 1.2

Proportional 0.02

Integral gain 0.00008

Tank cross-sectional area (A) 5

Fluid inflow rate (expected/manual and actual) 0.0394 0.0474

Inlet flow disturbance ON

Outlet valve errors

Manual control

Outlet valve (R)

Expected R 0.06

Actual R 0.05

Fig. 1. GUI for tank level control.

Section 2 will give a brief review of the MATLAB GUI environment. Section 3 introduces the learning outcomes the GUI is designed to reinforce and support and section 4 details the GUI design. The paper finishes with conclusions.

## 2. BACKGROUND ON USING MATLAB TO DEVELOP VIRTUAL LABORATORIES

A summary of how to use the MATLAB guide interface is given concisely for completeness only (e.g. see Rossiter (2012, 2016) for more). Type *guide* in the command window to open the editor.

- A palette of commonly used interactive buttons is included by dragging, positionning and sizing with the mouse. All the properties are visible in a window.
- Having designed the interface, MATLAB automatically builds a partner m-file with sub-functions for each interactive item. The GUI activates the associated subfunction whenever the button is selected by the user; the programmer enters the relevant code, accessing all of MATLAB's capabilities as required.

Animation in MATLAB is straightforward using loops with pauses and editting figure object properties (colour, position, etc). As long as the pictures are respresentative enough (even cartoon like), they are nevertheless good enough to convey a clear mental image to the student; certainly that is the impression his students have given the author. For example see figure 1 for a tank level GUI.

## 3. TECHNICAL LEARNING OUTCOMES FOR INTRODUCTORY CONTROL COURSES

This section will summarise the learning outcomes to be tackled in the proposed virtual laboratory/learning resource. The context is an introductory control course which also includes dynamics. For an introductory course, some core items will be:

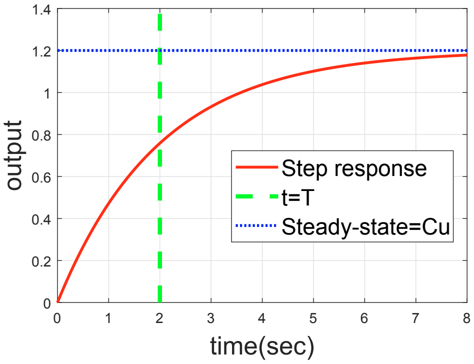


Fig. 2. Finding time constant and gain from a 1st order step response.

(1) Characterisation of 1st order models through time constant $T$ and steady-state gain $C$.
(2) Characterisation of 2nd order responses through damping ratio $\zeta$, time constant, decay rate/overshoot and natural frequency.
(3) Definition of open- and closed-loop poles.
(4) Impact of proportional design on system behaviour (e.g. poles, time constant, steady-state, offset).
(5) Impact of PI design on system behaviour.
(6) Introduction to systematic design of PI compensators.

This list is not comprehensive because for ease of use, a single GUI is limited in what it can display and the number of questions it can require a student to answer. One might argue that the GUI presented here is already overloaded, although the intention was to include an entire control design process, from modelling through to PI design and evaluation.

### 3.1 First order modelling and dynamics

A first order model in standard from is given as:

$$T\frac{dx(t)}{dt}+x(t)=Cu \tag{1}$$

where $u=u(t)$ is assumed constant. A simple modelling technique is to estimate the time constaint and gain from a step response as shown in figure 2, for example using 63% rise for the time constant $T$ estimation and noting that the steady-state is given by $Cu$.

### 3.2 Second order modelling and dynamics

A second order model in standard from is given as:

$$\frac{d^2x}{dt^2}+2\zeta w_n\frac{dx}{dt}+w_n^2x=Cw_n^2u \tag{2}$$

A simple modelling technique could be to estimate the gain $C$, damping ratio $\zeta$ and natural frequency $w_n$ from a step response as shown in figure 3, although typically one would give the model parameters and ask the student to determine the decay rate, damping ratio and poles. Students would be expected to know (or derive) that the decay rate is given by $e^{-2\zeta\pi/\sqrt{1-\zeta^2}}$ and the oscillation frequency is $w_n\sqrt{1-\zeta^2}$.

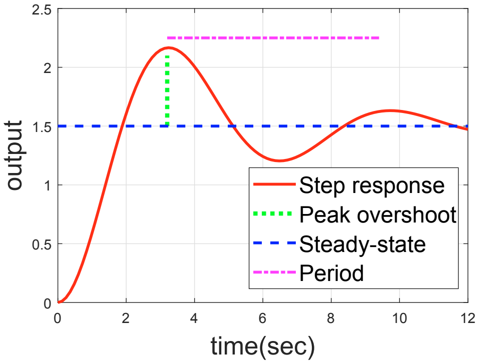


Fig. 3. Finding overshoot, period and gain from a 2nd order step response.

### 3.3 Pole positions and transfer functions

Students should be able to represent ODEs by equivalent transfer function models and identify the pole positions in the complex plane. They should also be able to discuss and analyse the expected behaviour for poles with non-zero imaginary parts and understand the significance of the left half plane. Hence for models (1,2) we have:

$$G_1=\frac{C}{Ts+1};\quad G_2=\frac{Cw_n^2}{s^2+2\zeta w_ns+w_n^2} \tag{3}$$

with corresponding poles at $-1/T$ and $-\zeta w_n \pm jw_n\sqrt{1-\zeta^2}$ respectively.

### 3.4 Closed-loop analysis

The basic skills needed for an introductory course link to an understanding of behaviours and the impact of compensation on behaviour. Typically behaviour is quantified using characteristics such as: i) poles; ii) time constants; iii) damping and overshoot; iv) offset (oftens as %); v) rise-time; vi) settling time.

In order to assess the impact of compensation on behaviour, students need to be able to create closed-loop transfer functions which for a typical control loop containing compensator $M(s)$ and system $G(s)$ would reduce to:

$$G_c=\frac{GM}{1+GM} \tag{4}$$

For PI design, compensators would be given as:

$$M(s)=K_p+\frac{K_i}{s} \tag{5}$$

hence, students would design or be given a compensator $M(s)$ and use this with (4) to find the closed-loop transfer function and then use this in conjunction with the skills from sections 3.1,3.2,3.3 to do some closed-loop analysis.

### 3.5 PI design

In an introductory course students will be given simple guidance on PI design. The presented GUI here uses design rules which work well for 1st order systems (1) and are summarised as follows where the design specification is for the closed-loop to be a factor $R$ faster than the open loop.

$$M(s)=\frac{R}{C}+\frac{R}{TCs}=\frac{R}{CT}\frac{Ts+1}{s} \tag{6}$$

## 4. GUI DESIGN FOR SELF-ASSESSMENT

The GUI is designed to capture and assess student competence in the core skills, beginning from modelling, through closed-loop analysis and finally, for PI design. Each time a random and unknown $G(s)$ is selected by the system. There are two separate GUIs because to meet both sets of objectives on a single screen makes fonts, figures and boxes too small and the interface too complex for the desired simple use.

### *4.1 GUI layout with feedback provided*

The GUI (Figure 4a) is designed with separate panels for each core skill so students and staff can organise the learning outcomes. Each panel has a *Go* button and a *check* button. The *Go* button is used to update the figure and closed-loop transfer functions shown in the red boxes. The *check* button assesses their numerical entries and gives a hint if incorrect. Students can still use the command window of MATLAB and thus perform computations there such as *tf.m, feedback.m, pzmap.m* before entering values into the GUI.

- The top left panel does modelling of a 1st order system from a step response. Students estimate $T$ and $C$. When the students select 'check' in this panel, the figure updates to overlay the step response of their model (denoted *user* $G$) with the actual system so they can verify whether the two are the same or not. Feedback also appears in the relevant boxes against each numerical computation. This panel also asks them for the expected value of $K_p$, based on the guidance in eqn.(6).
- The bottom left panel checks analysis of closed-loop behaviour with only proportional feedback, and thus some offset is expected. The answers are checked using the *user* $G$ from panel 1 (shown in the purple box below the 2nd panel) so even if students have this wrong, it will assess their working. Of course ideally students would not progress past panel 1 until their $G(s)$ parameters were correct. Students need to form a closed-loop transfer function for this panel, but the expected values are given in the red box of panel 1, when they select 'Go', to assist them.
- The middle bottom panel checks the student PI design against the correct design of (6), again using the *user* $G$ and also potentially introduces some 2nd order dynamics, as given in section 3.2, although if their steps in the first 2 panels are correct and their PI is correct, the closed-loop behaviour will be 1st order. Students need to form a closed-loop transfer function for this panel, but the expected values are given in the red box, when they select 'Go', to assist them.
- The panel on the far right is focussed on 2nd order dynamics which will ensue for a more general PI design, and thus allows students to check their computation of decay rates, damping ratios and complex poles and implied time constants. Students need to form a closed-loop transfer function for this panel, but again the expected values are updated in the red box when they select 'Go' to assist them and for checking against their working in the main command window.

### *4.2 GUI without feedback*

If a staff member would like to assess students without giving interim feedback, it is straightforward to prune the GUI down to return only correct/incorrect against each criteria. For completeness an example is given in figure 4b, where the reader will note the *Check Answer* panels and red boxes containing the implied closed-loop transfer functions have been removed and now simply a mark is returned.

## 5. STUDENT ENGAGEMENT

While GUIs can be wonderful tools for conscientious students who will readily use them to support their learning, less conscientious students may use them less, or even not at all unless there is a clear requirement to do so. Hence, in the author's experience an effective, but not the only solution, is to give the students situations where the GUI use is compulsory. One simple trick is to include use of the GUIs in tutorial sheets or assignments, even if in a somewhat trivial fashion, e.g.:

- Preamble .... With the tank GUI, what happens to the closed-loop behaviour if the PI is unchanged and the tank volume is increased?
- Preamble ... With the heat exchanger, what is the impact on disturbance rejection of increasing the proportional gain?
- Preamble ... What is the closed-loop time constant?

The author has also found their use in lectures can be entertaining up to a point, for example of you deliberately cause an over flow or other catastrophe. The animation capability is particularly helpful here as you cannot easily, or at all, being meaningful equipment into a lecture theatre. The slightly cartoon nature alongside meaningful scenarios helps students engage with the importance of the topic and core issues. Nevertheless, a core point is that demonstration within a lecture is easy given the GUIs are MATLAB based and thus, the same would apply here: the staff member would demonstrate how the GUI works so students are more comfortable using thereafter.

It is also important that students have simple access to the GUI files. The author provides access via a University shared folder so students can run directly from that folder within the University network, or, if they prefer, copy the files to their own machine. nevertheless, to avoid possible corruption, usually only the p-code is provided to the students. Some student quotes on efficacy:

- *MATLAB GUIs all help develop understanding*
- *Numerous GUIs and praticals helped me understand module.*

## 6. CONCLUSION

This paper begins from the assumption that virtual laboratories play a useful role in a modern engineering education. They give students freedom to experiment in their own time without access restrictions. However, having established that MATLAB GUIs are a simple and cost effective way of developing VL, it also becomes clear that these can be used to develop more general learning resources to support independent learning and self-assessment. Consequently, the proposal in this paper is for a GUI that supports learning of core mathematical skills linked to a first course in modelling and control. The GUI allows students to test their basic competence in calculating core measures of performance such as time constant, poles, overshoots, decay rates and so forth, and also, simultaneously, their ability to properly tune and assess the performance of a PI compensator.

By using the GUI environment, it is possible to include hints linked to student responses and thus help them move towards the correct answer. The hints provided in this GUI are relatively concise and will be assessed through student feedback and student conversations, but in principle one can activate MATLAB to give much more comprehensive hints should the lecturer find that desireable. This requires only augmentation of the relevant call back subfunction within the m-file: decide what feedback you want to give based on an analysis of the answer the student has given. The author also provides a guide to the GUI file which students can use alongside. This explains the required background to the computations and the working of the GUI. However, it is recommended that 5 minutes of a lecture or tutorial class is used to give a live demonstration to give students confidence in getting started.

## REFERENCES

Abdulwahed, M. (2010), Towards enhancing laboratory education by the development and evaluation of the trilab concept, PhD Thesis, University of Loughborough

Beal, L.D.R., Hill, D., Martin, R.A., Hedengren, J.D. (2018). GEKKO Optimization Suite, Processes, 6, 8, doi: 10.3390/pr6080106

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

Chen, X., G. Song, and Y. Zhang (2010). Virtual and remote laboratory development: A review, In *Earth and Space, Engineering, Science, Construction, and Operations in Challenging Environments*, pages 3843–3852, doi: 10.1061/41096(366)368.

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.

The DIAMOND building, http://www.sheffield.ac.uk/diamond

Dormido, S., H. Vargas, J. Sanchez (2012). AutomatL@bs Consortium: A Spanish Network of Web-Based Labs for Control Engineering Education, Internet Accessible Remote Laboratories: Scalable E-Learning Tools for Engineering and Science Discipline, 11, 206-225, A. Azad, M. E. Auer, V. J. Harward (Ed), IGI Global.

Fabregas, E., G. Farias, S. Dormido-Canto, S. Dormido, and F. Esquembre (2011). Developing a remote laboratory for engineering education. Computers & Education 57:1686-1697.

Farias, G., Munoz de la Pena, D., Gomez-Estern, F., De la Torre, L., Sanchez, C., and Dormido, S. (2016). Adding automatic evaluation to interactive virtual labs. Interactive Learning Environments, 24(7), 1456-1476.

Goodwin, G. (2010). Virtual laboratories for control systems design. http://www.virtual-laboratories.com/(last checked 1/9/10).

Goodwin G.C., A. M. Medioli, W. Sher, L. B. Vlacic, and J. S. Welsh (2011). Emulation-based virtual laboratories: A low-cost alternative to physical experiments in control engineering education, IEEE Transactions on Education, 54:48-55.

Guzman, J., K. Astrom, S. Dormido, T. Hagglund and Y. Piguet (2006). Interactive learning modules for pid Control, IFAC symposium on Advances in Control Education.

Hedengren, J.D. (2019). Temperature Control Lab Kit, http://apmonitor.com/heat.htm

Munoz de la Pena, D., Gomez-Estern, F., and Dormido, S. (2012). A new Internet tool for automatic evaluation in control systems and programming. Computers & Education, 59(2), 535550.

Olivier, G. W. J., Herson, K., and Sosabowski, M. H. (2001). WebMark a fully automated method of submission, assessment, grading, and commentary for laboratory practical scripts. Journal of Chemical Education, 78, 16991703.

Perez, J., S. Dormido and L. Vlacic (2011). Enhancing student learning: On-line interactive laboratory for modelling of real world control system applications, IFAC world congress, pp.7268-7273.

Y. Qiao, G. Liu, G. Zheng and C. Luo (2012). Design and realization of networked control experiments in a web-based laboratory, Proc. UKACC.

Rossiter, J.A., Y. Baradaranshokouhi, I. Lilley and C. Bacon (2011). Developing web accessible laboratories for introductory systems and control using student projects, IFAC world congress.

Rossiter, J.A. (2012). Using MATLAB to create cheap and accessible virtual laboratories, ISEE, 2012.

Rossiter, J.A. (2016). Low production cost virtual modelling and control laboratories for chemical engineering students, ACE 2016.

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

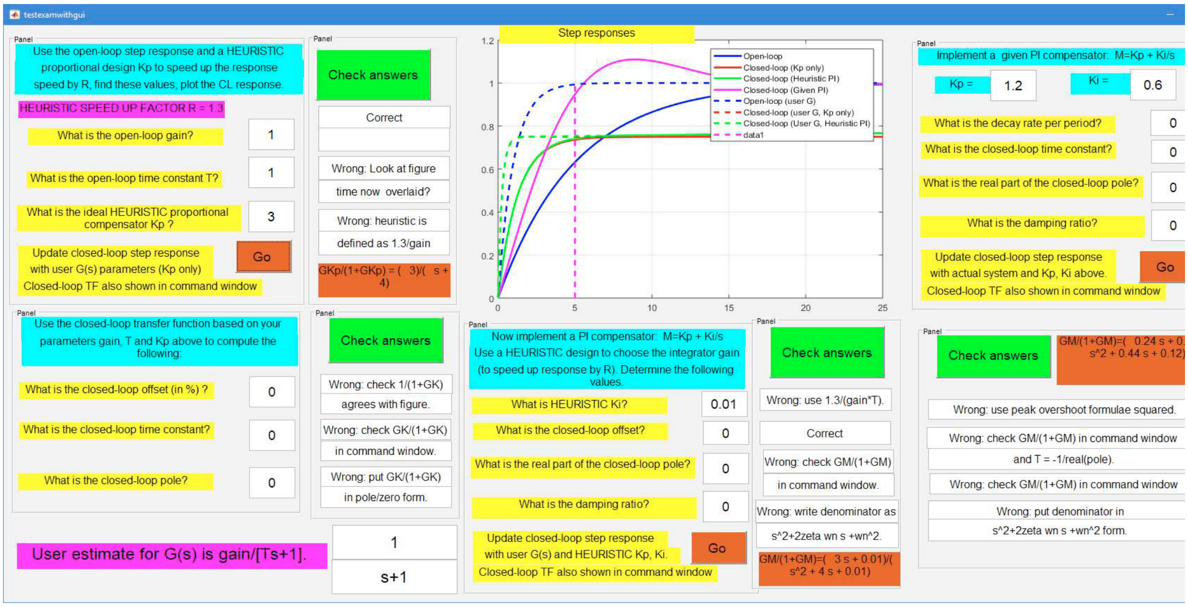
Rossiter, J.A., Pasik-Duncan, B., Dormido, S., Vlacic, L., Jones, B. and Murray, R. (2018). Good Practice in Control Education, European Journal of Engineering Education, http://dx.doi.org/10.1080/03043797.2018.1428530

Rossiter, J.A., Pope, S.A., Jones, B. Ll and Hedengren, J.D. (2019). Evaluation and demonstration of take home laboratory kit, ACE 2019.

Stark, B., Li, Z., Smith, B. and Chen, Y. (2013). Take-Home Mechatronics Control Labs: A Low-Cost Personal Solution and Educational Assessment, Proceedings ASME.

The STEMLAB building, http://www.lboro.ac.uk/departments/meme/undergraduate/student-facilities/stemlab/

Vargas, H., J. Sanchez, C. A. Jara, F. A. Candelas, F. Torres, S. Dormido (2011). A Network of Automatic Control Web-based Laboratories, IEEE Transactions on Learning Technologies, 4, 3, 197-208.



Use the open-loop step response and a HEURISTIC proportional design Kp to speed up the response speed by R, find these values, plot the CL response.

HEURISTIC SPEED UP FACTOR R = 1.3

What is the open-loop gain? 1

What is the open-loop time constant T? 1

What is the ideal HEURISTIC proportional compensator Kp ? 3

Update closed-loop step response with user G(s) parameters (Kp only) Closed-loop TF also shown in command window

Correct

Wrong: Look at figure time now overlaid?

Wrong: heuristic is defined as 1.3/gain

GKp/(1+GKp) = ( 3)/( s + 4)

Use the closed-loop transfer function based on your parameters gain, T and Kp above to compute the following:

What is the closed-loop offset (in %) ? 0

What is the closed-loop time constant? 0

What is the closed-loop pole? 0

Wrong: check 1/(1+GK) agrees with figure.

Wrong: check GK/(1+GK) in command window.

Wrong: put GK/(1+GK) in pole/zero form.

User estimate for G(s) is gain/[Ts+1]. 1 / s+1

Now implement a PI compensator: M=Kp + Ki/s Use a HEURISTIC design to choose the integrator gain (to speed up response by R). Determine the following values.

What is HEURISTIC Ki? 0.01

What is the closed-loop offset? 0

What is the real part of the closed-loop pole? 0

What is the damping ratio? 0

Update closed-loop step response with user G(s) and HEURISTIC Kp, Ki. Closed-loop TF also shown in command window

Wrong: use 1.3/(gain*T).

Correct

Wrong: check GM/(1+GM) in command window.

Wrong: write denominator as s^2+2zeta wn s +wn^2.

GM/(1+GM)=( 3 s + 0.01)/( s^2 + 4 s + 0.01)

Implement a given PI compensator: M=Kp + Ki/s

Kp = 1.2 Ki = 0.6

What is the decay rate per period? 0

What is the closed-loop time constant? 0

What is the real part of the closed-loop pole? 0

What is the damping ratio? 0

Update closed-loop step response with actual system and Kp, Ki above. Closed-loop TF also shown in command window

GM/(1+GM)=( 0.24 s + 0. s^2 + 0.44 s + 0.12)

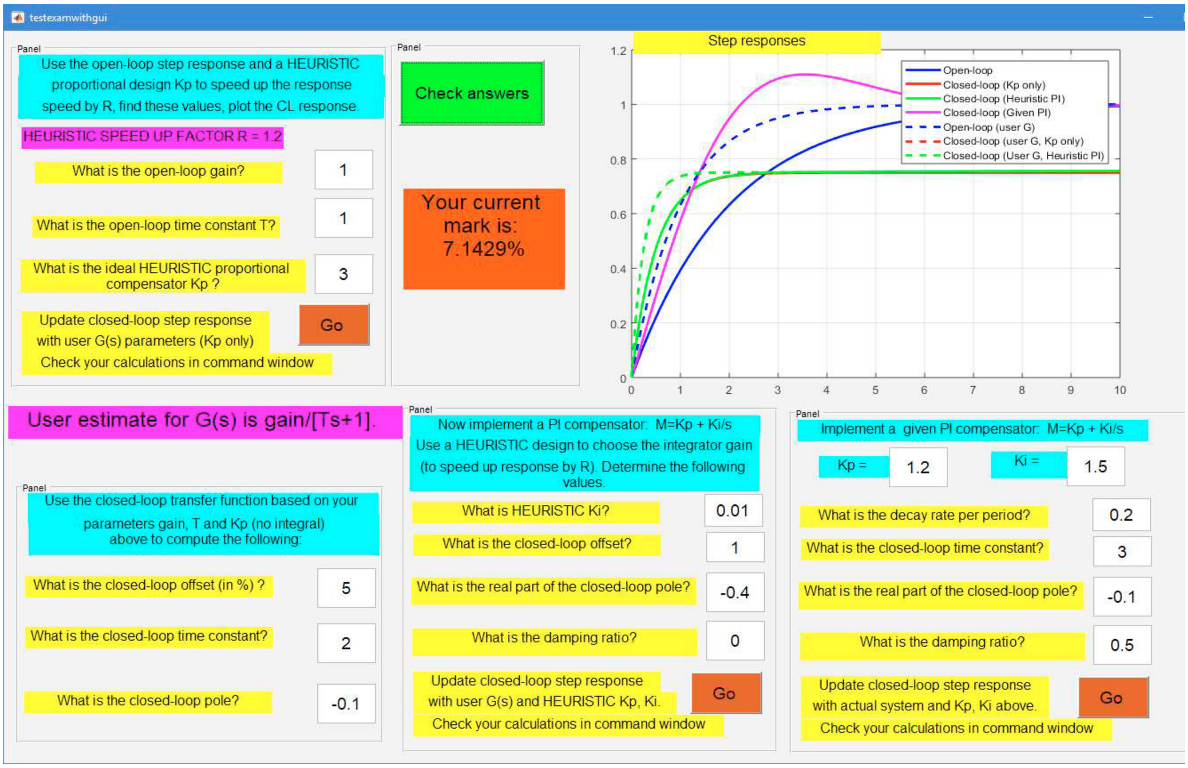
Wrong: use peak overshoot formulae squared.

Wrong: check GM/(1+GM) in command window and T = -1/real(pole).

Wrong: check GM/(1+GM) in command window

Wrong: put denominator in s^2+2zeta wn s +wn^2 form.

(a) Interface with feedback assistance.



Use the open-loop step response and a HEURISTIC proportional design Kp to speed up the response speed by R, find these values, plot the CL response.

HEURISTIC SPEED UP FACTOR R = 1.2

What is the open-loop gain? 1

What is the open-loop time constant T? 1

What is the ideal HEURISTIC proportional compensator Kp ? 3

Update closed-loop step response with user G(s) parameters (Kp only) Check your calculations in command window

Your current mark is: 7.1429%

User estimate for G(s) is gain/[Ts+1].

Use the closed-loop transfer function based on your parameters gain, T and Kp (no integral) above to compute the following:

What is the closed-loop offset (in %) ? 5

What is the closed-loop time constant? 2

What is the closed-loop pole? -0.1

Now implement a PI compensator: M=Kp + Ki/s Use a HEURISTIC design to choose the integrator gain (to speed up response by R). Determine the following values.

What is HEURISTIC Ki? 0.01

What is the closed-loop offset? 1

What is the real part of the closed-loop pole? -0.4

What is the damping ratio? 0

Update closed-loop step response with user G(s) and HEURISTIC Kp, Ki. Check your calculations in command window

Implement a given PI compensator: M=Kp + Ki/s

Kp = 1.2 Ki = 1.5

What is the decay rate per period? 0.2

What is the closed-loop time constant? 3

What is the real part of the closed-loop pole? -0.1

What is the damping ratio? 0.5

Update closed-loop step response with actual system and Kp, Ki above. Check your calculations in command window

(b) Interface with marking only.

Fig. 4. Interfaces for the GUIs to assess 1st and 2nd order dynamics and PI design.