

A review of Matlab's SISOTOOL; features and contributions to Control education

Chee Pin Tan** Kok Soo Teoh* Lim Jen Nee Jones*

* *School of Engineering, Monash University, Jalan Lagoon Selatan, Bandar Sunway, 46150 Selangor Darul Ehsan, Malaysia*

** *Engineering Department, Leicester University, University Road, LE1 7RH UK*

Abstract: 'SISOTOOL' is one of the latest features in the Control System Toolbox in Matlab that enables software-based controller design for single-input-single-output (SISO) systems. This paper provides a review of SISOTOOL, in particular its features that contribute to Control education, which will be useful to the academic community in teaching Control at undergraduate level.

Keywords: Control Education

1. INTRODUCTION

At undergraduate level, controller design for single-input-single-output (SISO) systems would normally involve the earlier analysis tools in Control; in particular root locus, or frequency response plots (most popularly, the Bode plot). As controller design is normally an iterative process, it will require re-drawing of those analysis plots, which can be tedious and time-consuming. For undergraduates who would have just learnt Bode plots and root locus very recently, this can be particularly frustrating and discouraging, and could therefore impede their learning. 'SISOTOOL' is a graphical user interface (GUI) in Matlab that was developed by the Mathworks to assist in the process of designing controllers for SISO systems. The iterative process of re-drawing the root locus or Bode plots is eliminated, as SISOTOOL updates (in real-time) those plots as the controller structure or parameters are modified. This paper provides a review of SISOTOOL from a user point of view, highlighting its main features in particular its contribution to the development and improvement of Control education. Though this product has been in existence for a few years, to date there has not been a review of it, with the exception of a quick mention by Guvenc & Guvenc (2004). This paper seeks to provide a review of SISOTOOL and its contribution to the enhancement of Control education.

2. SISOTOOL IN THE DESIGN PROCESS

2.1 *Decide on controller architecture*

The first step in controller design is to choose the architecture of the controller, whether it is to be in the feedforward or feedback loop. SISOTOOL allows the user to specify the controller configuration.

* Correspondence to: School of Engineering, Monash University, Jalan Lagoon Selatan, Bandar Sunway, 46150 Selangor DE, Malaysia, tan.chee.pin@eng.monash.edu.my

2.2 *Decide on controller structure*

After the architecture has been decided, the user needs to specify the structure of the controller, whether to use a lead, lag, PI, PD, PID controller, or even any other controllers whose number of poles and zeros do not match those mentioned. Following that, the poles and zeros need to be added to the design plots. SISOTOOL will instantaneously show how the design plots change as the poles/zeros are added in. This will enable the user to immediately check if the objectives (for example, if the root locus can pass through the desired poles) can be achieved using that particular controller structure.

SISOTOOL provides the option of inserting and adjusting the poles and zeros either graphically or numerically; each mode having its own advantages. Inserting them numerically is advantageous when precise values are wanted (for example a pole at the origin for a controller with Integral term). By inserting them (and also moving them) graphically the user can perform adjustments very quickly to see which controller configuration will be able to let the root locus approximately pass through the desired poles.

In SISOTOOL, there are 3 design plots to choose from; root locus, Bode and Nichols. It is also possible to view multiple design plots for a system so that the user can verify that the controller structure can satisfy all objectives especially in the case of objectives in different domains; for example if there is a specification of percentage overshoot (in root locus) and gain margin (in frequency response). Whenever any controller parameters are changed, all design plots will also be correspondingly varied.

2.3 *Calculate controller parameters/gains*

Following the determination of the controller structure, the numerical values of the controller need to be calculated. The root locus and Bode plot are updated instantaneously each time there is a change in poles/zeros or even the gain.

This feature is particularly useful in design by Bode plot where there is (intelligent) guesswork involved, especially in the design of lead/lag compensators where the user has to specify (guess) an extra allowance for the Phase Margin. This guess-work would sometimes result in several design iterations which is virtually eliminated with this real-time update feature.

In controller design using root locus, one can move any Closed-Loop pole (hence changing the Closed-Loop gain) and the other poles will move correspondingly as the gain changes. This makes it very easy to check that the desired poles are dominant.

In addition to the real-time update feature, SISOTOOL also aids design by allowing users to set design constraints on their design plots. For the root locus plot, the user can specify the maximum allowable overshoot and settling time, which are real physical parameters. Then SISOTOOL will show which region on the root locus the dominant poles must be in so that the specifications can be specified. For the frequency response plots, users can specify the minimum required phase and gain margins, and SISOTOOL will show the visible bounds on the design plot.

2.4 Testing of controller

After the controller has been designed, it needs to be tested, as the mathematical design is only an 'intelligent guess' of what the controller should be. For example, even though the dominant poles lie in the desired regions, other closed-loop poles/zeros can still affect the response adversely.

SISOTOOL also provides the time response of various parameters that are useful to Control engineers, for example the time response of the output and the time response of the control input. The user can choose which the response variable of interest, and also choose whether a step or impulse response is required.

In addition, SISOTOOL can also update the time responses in real-time if there are any changes or modifications to the design. This is particularly useful if the initial design of the controller does not specify specifications.

In the event when a closed-loop zero causes the response to be too fast or to have overshoot that is too high, a pre-filter would be required. SISOTOOL also makes provision for the requirement of a pre-filter. The user can opt to have a pre-filter and also specify the poles of the pre-filter. Consequently, the time response of the system will also be automatically updated.

3. DISCUSSION

The process of designing controllers could be very tedious due to the requirement of possible iterative designs. Even though it is extremely useful for students to firstly know the concept and drawing steps of the design plots (root locus and frequency response), the requirement to draw those design plots could make the controller design process very long. In addition, iterative re-designs can occur at the design stage (for complex systems with many poles/zeros)

and at the stage of testing the controller. In Control education, this could result in students making mistakes along the way, which could be very frustrating and discouraging. Consequently, students may be impeded in their learning.

SISOTOOL through its real-time update feature virtually eliminates this problem. If a student does sufficiently understand the concepts of the design plots, he/she could easily design a controller from the start to the end without much of a problem and quite quickly too.

Therefore, it can be concluded that SISOTOOL does enhance Control education mainly by its feature of real-time update.

4. CONCLUSION

This paper has provided a review of the product SISOTOOL in Matlab, highlighting its main features and explaining how it contributes to the enhancement Control education. It can be concluded that SISOTOOL cuts down the time and effort for the design process with its real-time update feature; this has simplified the process of controller design, which makes it more appealing to students, and hence, enhances Control education.

REFERENCES

- L. Guvenc and B.A. Guvenc *Design projects on automotive controls - developing an automation lab for senior projects*. IEEE Control Systems Magazine, vol. 24, p.p. 92 - 42.