

Documentation for

PI/PID Design GUI for MATLAB

Copyright (C) 2002-2006: Juergen Hahn

Department of Chemical Engineering

Texas A&M University

College Station, TX 77843-3122

USA

hahn@tamu.edu

Introduction

Controllers of PID type are by far the most widely used controllers in the chemical process industries. However, in many cases the controllers are not properly tuned resulting in a significant loss of performance. In order to counter this, new and simple tuning rules have been derived for single-loop PID controllers. These rules can be automated in order to assist with the design of PI and PID controllers. This software implements existing PI and PID tuning rules for processes that are either given in transfer function form or where data for the input-output behavior of the process is available. The controller tuning and evaluation of the closed-loop performance is performed interactively using a graphical user interface in order to enhance ease of use.

Installation Instructions

- 1) Download the software.
- 2) Create a directory for the software.
- 3) Extract the files into the newly created directory.
- 4) Add the directory to the path in MATLAB.
- 5) From within MATLAB, type 'PID_Design' to start the GUI.

System Requirements

The software was tested using MATLAB 6.1 and the Optimization Toolbox 2.1.

How to use the software

After the GUI has been launched the main control window will appear on the screen. The window is shown in Figure 1 and serves as the main menu from which the other windows are started. The menu options are listed in the same order as they will have be called in a regular work session. This includes importing a model (from either a transfer function model or from data), designing a controller, and analyzing properties of the closed-loop system. The first three buttons will open windows where the action is performed. There is also a fourth menu option 'Quit' that will leave the program and close the main window.

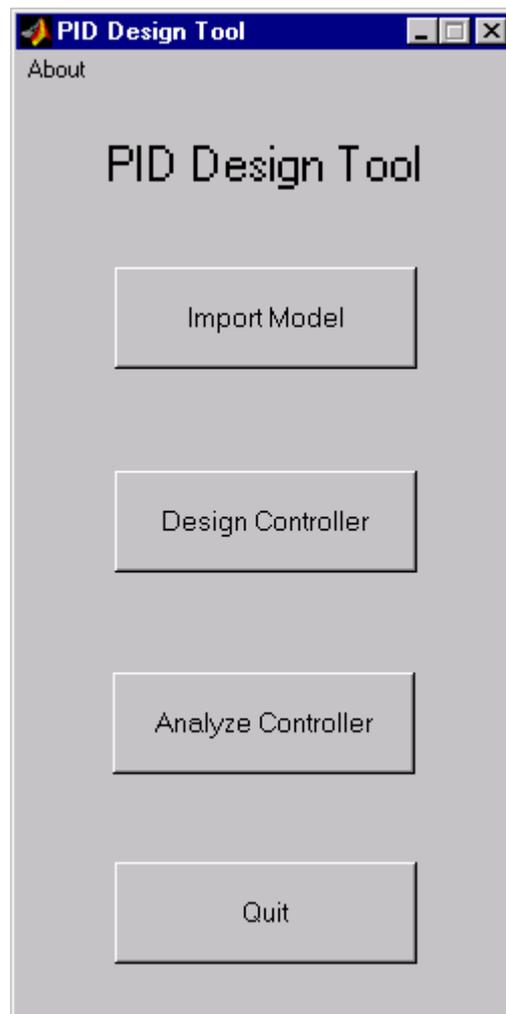


Figure 1: PID Design Tool main menu

Import Model window

This window (shown in Figure 2) enables the user to input a model for the system that is to be controlled. The model can be entered in transfer function form or data (time 't' and the output 'y') representing a unit step response of the system can be imported from the workspace. When the 'Import TF' button is pressed then the model parameters for the numerator and denominator polynomial as well as for the time delay are used for the model. If the 'Import Data' button is pressed then the variables that were chosen for 't' and 'y' are imported and used as the systems response. It should be noted that the values for 'y' have to correspond to the system output for a unit step input. If other forms of inputs were chosen for generating the data then 'y' needs to be pre-processed on the MATLAB workspace before it is imported. Pressing either button will disregard whatever actions have been taken for the "other" form of importing a model. Also, the window will be closed after the model has been imported.

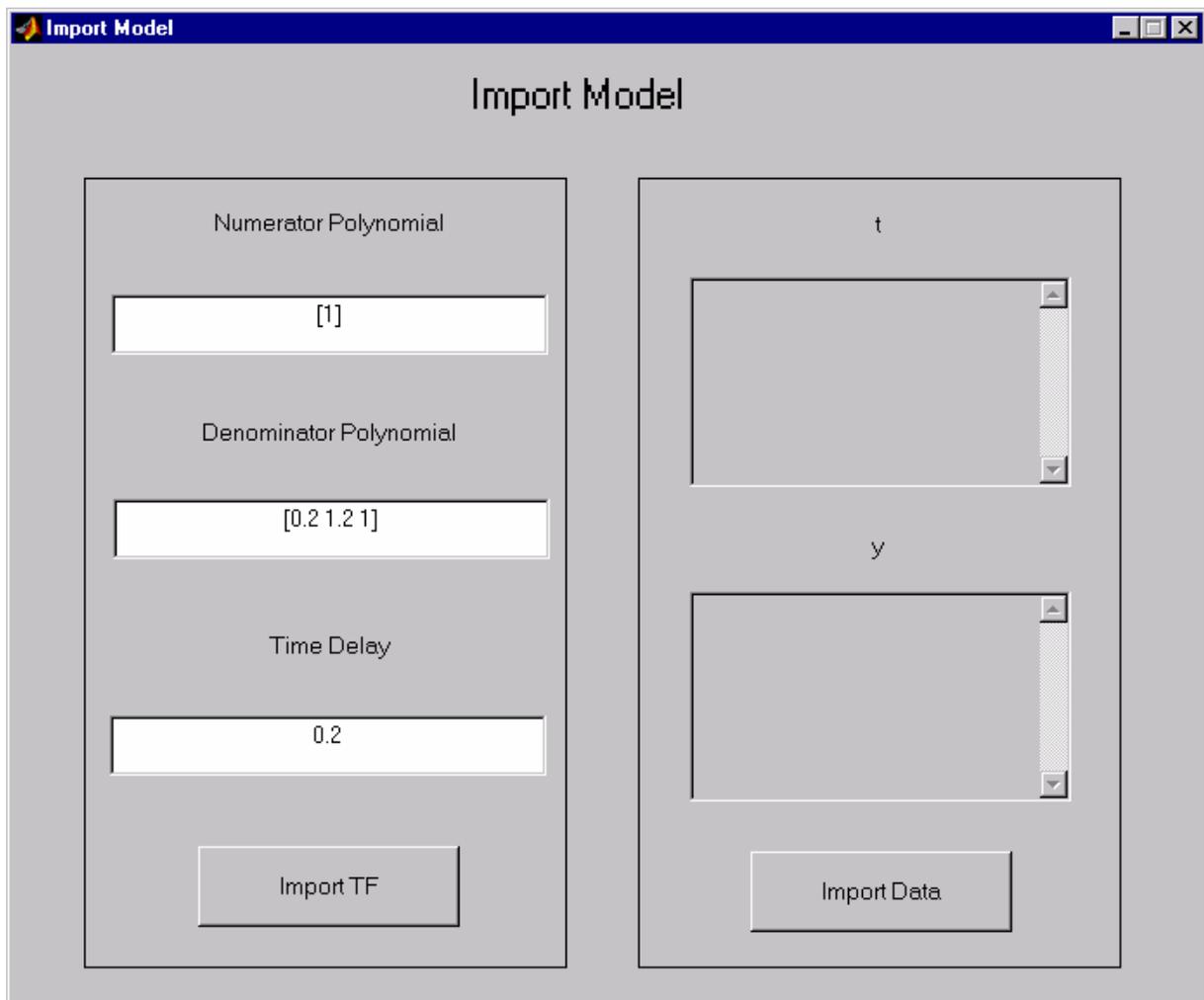


Figure 2: Import Model window

Controller Design window

This window represents the core component of the PID Design Tool. Here the model transfer function is simplified, the controller structure is chosen, the controller is designed, the controller can be tested using various impulse as well as step responses, and the controller can be retuned. This allows for an interactive controller tuning method, because it is trivial to change tuning parameters and immediately observe the effect that this change will have on the closed-loop system. Figure 3 shows the window as it will appear when it is started from the main menu.

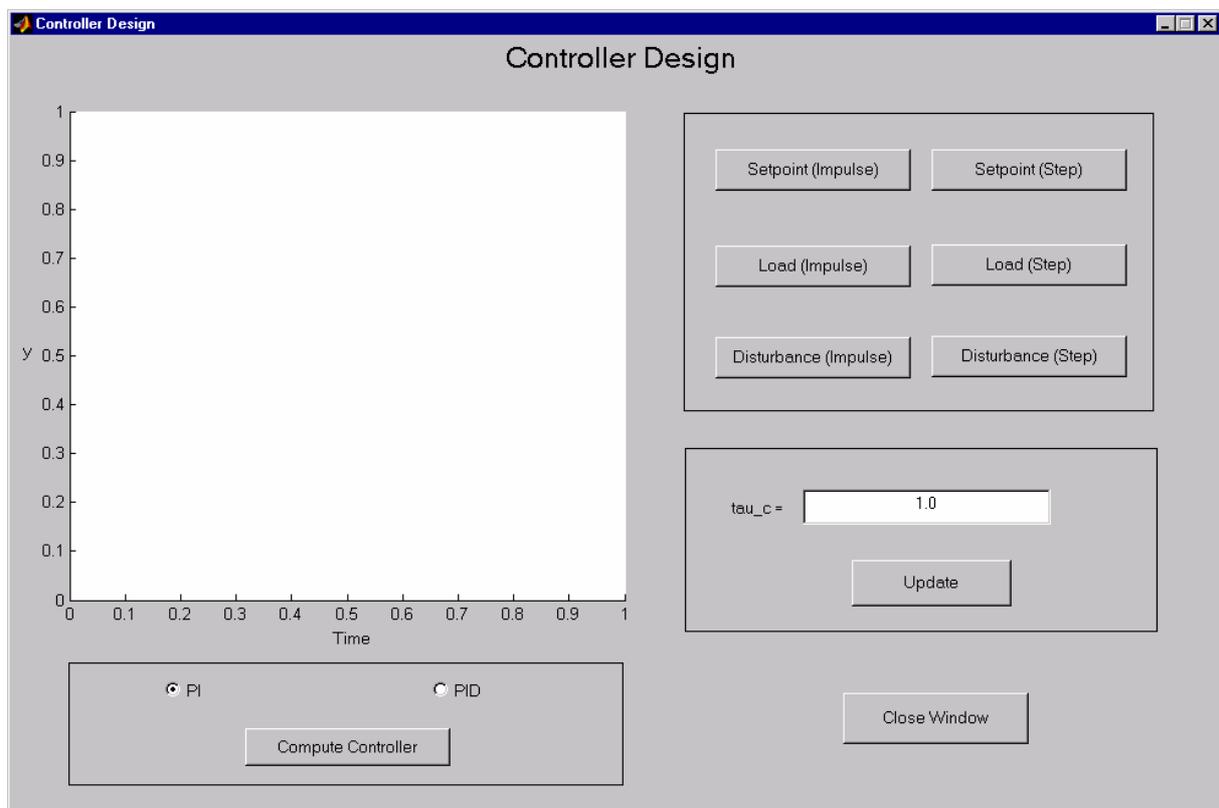


Figure 3: Controller Design window

In a first step the type of controller has to be chosen. The program allows the design of controllers of PI as well as of PID type. For a PI controller, a first order transfer function that best represents the imported model is determined, whereas for a PID controller, a second order transfer function that best approximates the imported model is computed. Since the tuning method that is implemented in this tool is based upon a modification of Skogestad's internal model control (IMC) tuning procedure, the interested reader is referred to [1] for further detail on how the controller parameters are determined.

Once the controller type has been decided upon the ‘Compute Controller’ button needs to be pressed. As a result of this, the program will determine a simplified model structure and also suggest a value for the controller tuning parameter τ_c (tau_c). This parameter represents the desired time constant of the closed-loop process. Therefore, a smaller value of τ_c will result in a more aggressive controller, whereas large values of τ_c can result in poor performance. Since this method is based upon a modification of IMC tuning, τ_c is the only one tuning parameter that can be independently chosen and the coefficients for the proportional, the integrating, and the derivative part are determined by the model structure and the selection of this tuning parameter. When the program has computed a controller, the data that was imported as well as the fitted function that was used for the controller design are plotted in the window. This allows the user to determine if a good approximation was determined. It also signals that the computation process for the controller has finished (this step is the most computationally intensive one and will take several seconds). Additionally, the value of the tuning parameter is displayed in the window as is shown in Figure 4.

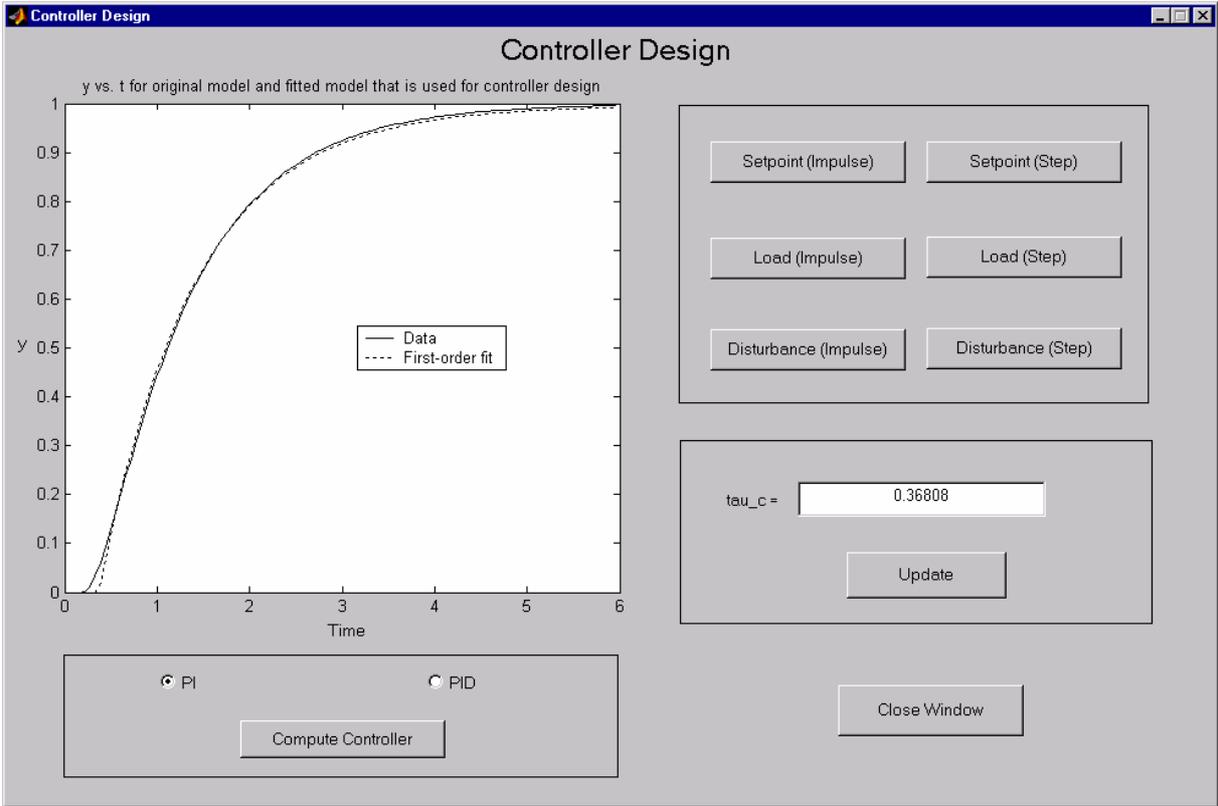


Figure 4: A simplified transfer function has been fitted and the controller has been designed. Note the suggested value for the tuning parameter tau_c

Additional information is reported on the MATLAB workspace. This includes the transfer function for the original model (keep in mind that when only data is imported, that this function is not known and instead the fitted transfer function is shown), the fitted transfer function, and a modified version of this fitted function that is used for controller design. This information also includes the coefficients of the P, I, and D parts of the controller.

Now that the controller has been computed, the closed-loop performance can be evaluated. There are six buttons that allow to compute the response to step as well as impulse changes in the setpoint, the load disturbance, and the output disturbance. The corresponding response will also be displayed in the window (Figure 5). Values for the various error criteria (IAE, ISE, ITAE) as well as for the evaluation of the amount of control energy (TV) are reported on the MATLAB workspace for each response of the system.

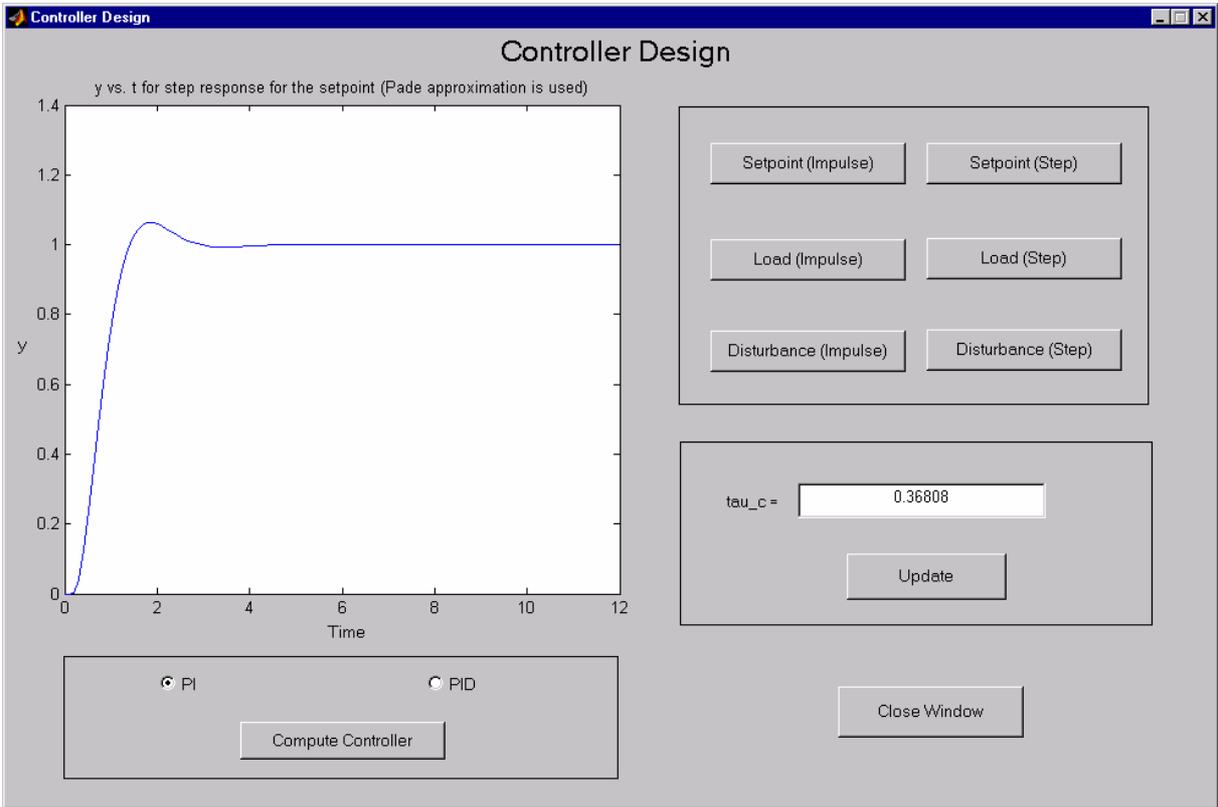


Figure 5: Step response for the setpoint for a PI controller

If the effect of different values of the tuning parameter for the closed-loop response should be investigated, then one can simply change the value for τ_c in the dialogue box and press the 'Update' button. This will recompute the controller tuning parameters and print them on the MATLAB workspace. Since this is only a retuning step and does not require that a new model structure is determined, it is performed very quickly. One can the compute and plot the

response of the system for this new tuning parameter. This procedure can be repeated several times and allows for interactive controller tuning and evaluation.

Another possibility is to change the controller type (from PI to PID or the other way around) and observe the effect that this change has on the closed-loop response as well as on the controller tuning parameter.

The 'Close Window' button closes the window and returns the program to the main menu window.

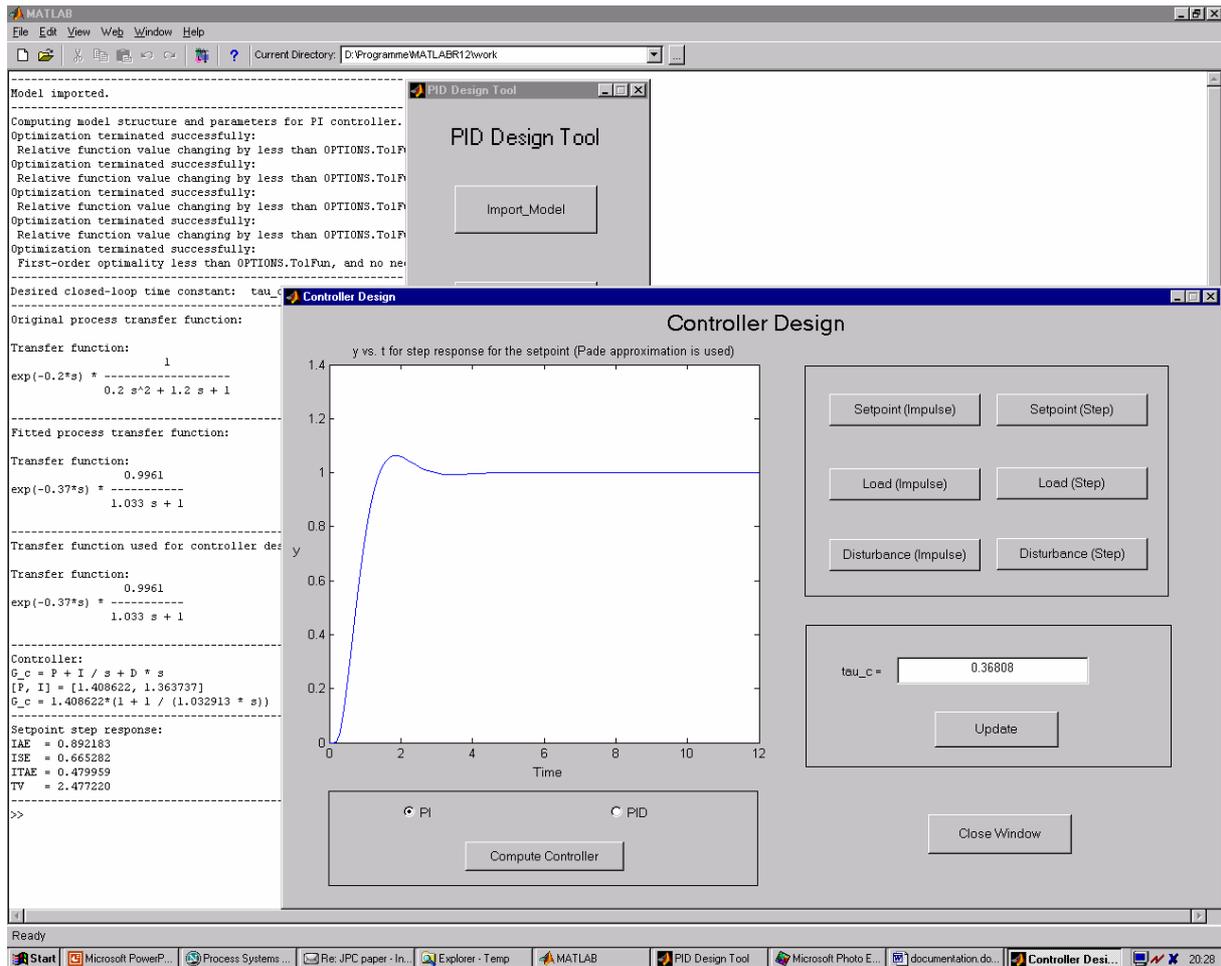


Figure 6: Controller Design window and information that appears on the MATLAB workspace

Figure 6 shows a work session using the PID Design Tool. A PI controller has been designed and the controller tuning parameter has been determined. The window shows the setpoint step response for the closed-loop system. In the background the information that is provided in the MATLAB workspace is shown.

Analyze Controller window

If additional information about the closed-loop system is required then this can be obtained in the 'Analyze Controller' window. Currently, this includes computation of the gain and phase margins, as well as drawing Bode and Nyquist plots (Figure 7).

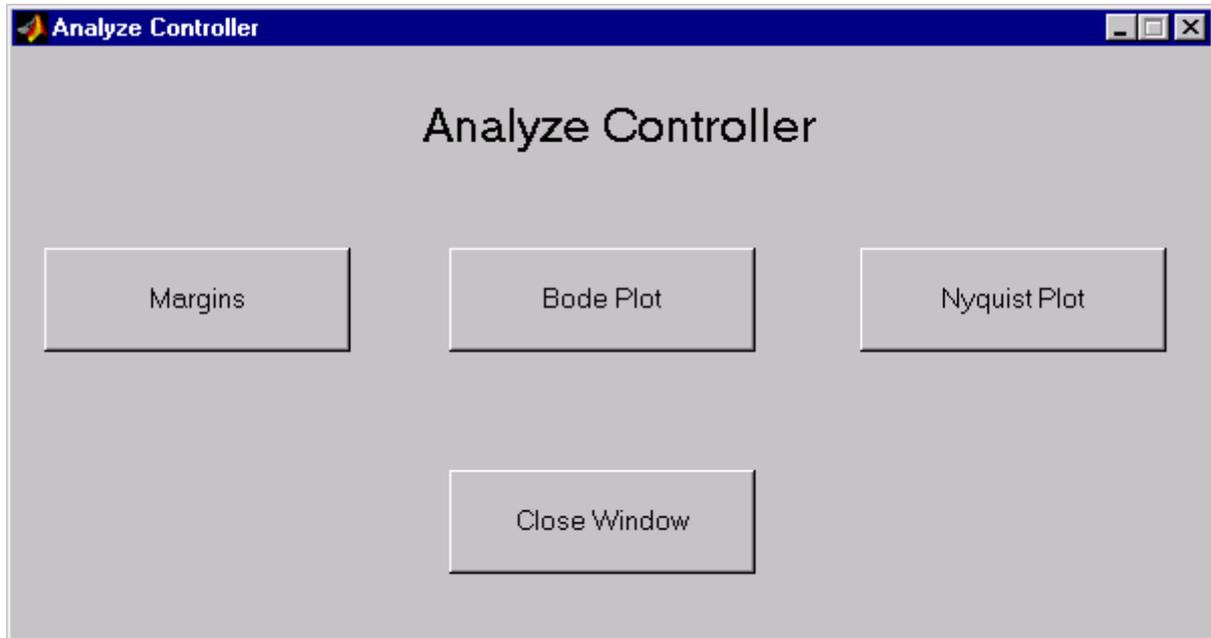


Figure 7: Analyze Controller window

Troubleshooting

The PID Design Tool has been tested on a variety of transfer function models and several models that were determined from process data. Generally, the performance of the closed-loop was very good and the design parameters were close to values that an experienced control engineer would compute by hand. However, there is always the possibility that the controller might not result in the desired performance. This can happen if the program is unable to determine a good value of the tuning parameter τ_c (the bounds for τ_c were chosen to be 0.05 and 200.0). If τ_c is determined to be at one of its bounds then it is likely that the controller does not have the desired performance. One possibility to address this problem is to retune the controller using the tool.

However, a more serious problem can result from numerical issues. If the model was imported as a transfer function then the program will first determine a step response from this model in order to fit a simpler model to it for the controller design. It can happen in a few special cases that the step response does not represent the whole dynamics that the process

exhibits and instead ends prematurely (this was found for some cases if one of the time constants was chosen to be much smaller than the dominant time constant → this is not necessarily a realistic situation since one has to ask if it is possible to identify such a process transfer function in the first place). For such a case a controller would be designed which is based-upon a transfer function that does not represent the dynamics of the real model. Due to this, it is likely that the closed-loop behavior will also not meet required performance criteria or might even become unstable. In order to draw attention to such a situation the user is presented with the step responses of the original transfer function as well as the fitted transfer function each time a new controller is implemented. If it turns out that these transfer functions do not represent the real behavior of the model (for example the dominant time constant could be 50.0 in a model, but the step response only goes up to a final time of 10.0), then one would have to generate a different data set in the MATLAB workspace and input this generated data set. Such a procedure should solve the above mentioned problem.

Reference

[1] S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*. Vol. 13, No. 4, pp. 291-309 (2003)