

WHEN YOU NEED TO BE SURE



# **CyFlex® Autotuning PID Loops**

**Version 3**

May 23, 2022

Developed by SGS North America, Inc.



## Version History

Version	Date	Revision Description
1	3/3/2020	Initial publication
2	8/23/2021	Added hyperlinked cross-reference to <a href="#">cyflex.com</a> usage help to autotune in <i>Section 3 Using Autotuning</i> on page 3.
3	5/23/2022	Updated hyperlinked cross-reference to <a href="#">cyflex.com</a> usage help to autotune in <i>Section 3 Using Autotuning</i> on page 3

## Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.  
Example: Select the `cmdapp-relVersion-buildVersion.zip` file....
- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.  
Example: **Type:** Click **Select Type** to display drop-down menu options.
- Cross-references are designated in Arial italics.  
Example: Refer to *Figure 1*...

## Related Documents

CyFlex manuals are available at <https://cyflex.com/>. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

## Table of Contents

<b>1</b>	<b>OVERVIEW</b> .....	<b>1</b>
1.1	BACKGROUND.....	1
<b>2</b>	<b>AUTOTUNING FUNCTIONS</b> .....	<b>2</b>
<b>3</b>	<b>USING AUTOTUNING</b> .....	<b>3</b>
3.1	USING THE RESULTS.....	7

## List of Figures

FIGURE 1: ENGINE CONTROL DISPLAY.....	3
FIGURE 2: SELECT A CONTROL DEVICE DIALOG.....	4
FIGURE 3: FILE TAIL PROGRESS DISPLAY.....	5
FIGURE 4: USER CONSOLE PROGRESS DISPLAY.....	6
FIGURE 5: PID GAIN VALUES.....	7
FIGURE 6: UPDATE LOGBOOK.....	8
FIGURE 7: UPDATE SPECIFICATION FILE.....	9

## 1 Overview

Although PID controllers are used in every test cell, expertise to select gains that will provide optimum performance for any given control loop is not widely known. The best gains may be different if the test article changes in size, flow rate, power level, or heat load. Gains may have to be tuned for several control loops each time a new test article is installed in a test cell. Therefore, a simple, fast, and reliable means for automatically tuning PID controllers is required. This document describes methods to select the gains that will provide optimum performance for any given control loop.

All features described in this document apply equally to user control loops.

### 1.1 Background

The autotuning functionality has been repaired and improved in the new engine and user control programs. It has not been modified in the old control task since no upgrades are being done to that program.

Although anyone can start the autotuning process, some site administrators will prefer that this command only be used by a fully qualified Application Engineer. At present, however, CyFlex does not have a means to prevent anyone from running the command. This presents a dilemma on just how to launch the function.

The new control programs are available in CyFlex 6.3.X. They require new/modified specification files. Review specification template files in `/cyflex/specs.def` if CyFlex 6.3.X is installed or refer to the following:

- [CyFlex User Control Loops](#)
- [CyFlex Engine Controls](#)

## 2 Autotuning Functions

The following describes the actions that take place when autotuning is invoked:

1. When first started, the autotuning function will take 150 readings of the feedback variable to establish the level of variability in the signal.
  - This will happen at the execution rate of the control loop so it may only take a few seconds at the fast rate or it may take over two minutes if the controller is being updated at once per second.
2. The program calculates two new outputs which are the present steady state output (X) times 1.2 and 0.8.
3. The program will then check to see if 1.2X and 0.8X are within the output bounds of the control device. This means that the starting point should be somewhere below 83% of the upper output bound.
  - A warning will inform the user if the steady state starting point should be moved.
4. The program will begin to toggle the output between the two new settings
  - It will hold the upper command setting until the feedback variable changes by more than 125% of the steady state noise.
    - As an example, if the maximum steady state speed error is 4 rpm at the start of the process, the program will maintain the upper output until the speed error exceeds 5 rpm. Then it will change the output to the lower limit until the error exceeds -5 rpm.
    - This will continue until a repeatable pattern of behavior is established which normally takes several cycles
    - The amount of time to reach a repeatable cycle depends on the responsiveness of the control loop. Temperature controllers are generally quite slow and may require patience during the autotuning process.
    - If the control loop is very responsive, the feedback variable may overshoot the 125% error threshold. This is normal and should be well controlled.
5. Once a steady oscillation pattern is established, the program will calculate the period and amplitude of the cycles and then use proprietary calculations to determine the best gains for the controller.
6. A stream of output will go to both the user console and to the control display GUI showing the progress of the program.

### 3 Using Autotuning

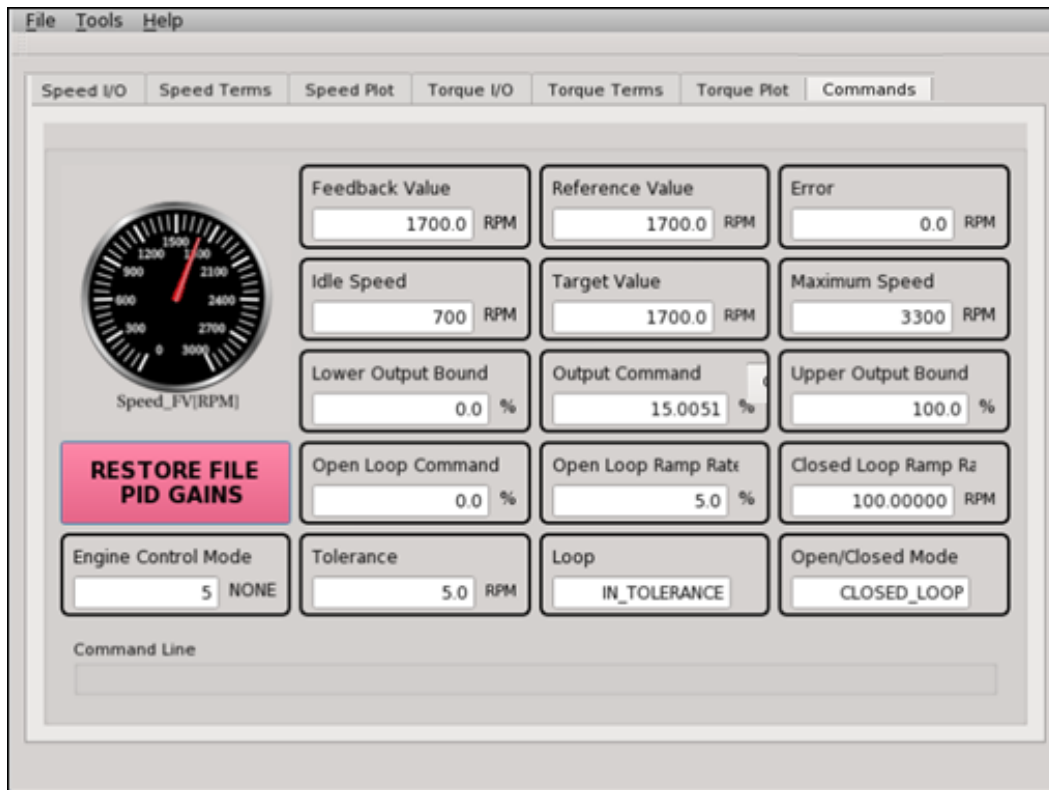
Use any of the following methods invoke the autotuning process:

- Command line; refer to cyflex.com usage help for [autotune](#).
- From within a command script
- The control display GUI; this is the preferred method and is described below.

The controller should be in closed loop and at a stable operating point before starting.

1. Select the **Commands** tab from the control display (engine or user) as in *Figure 1*.

*Figure 1: Engine Control Display*



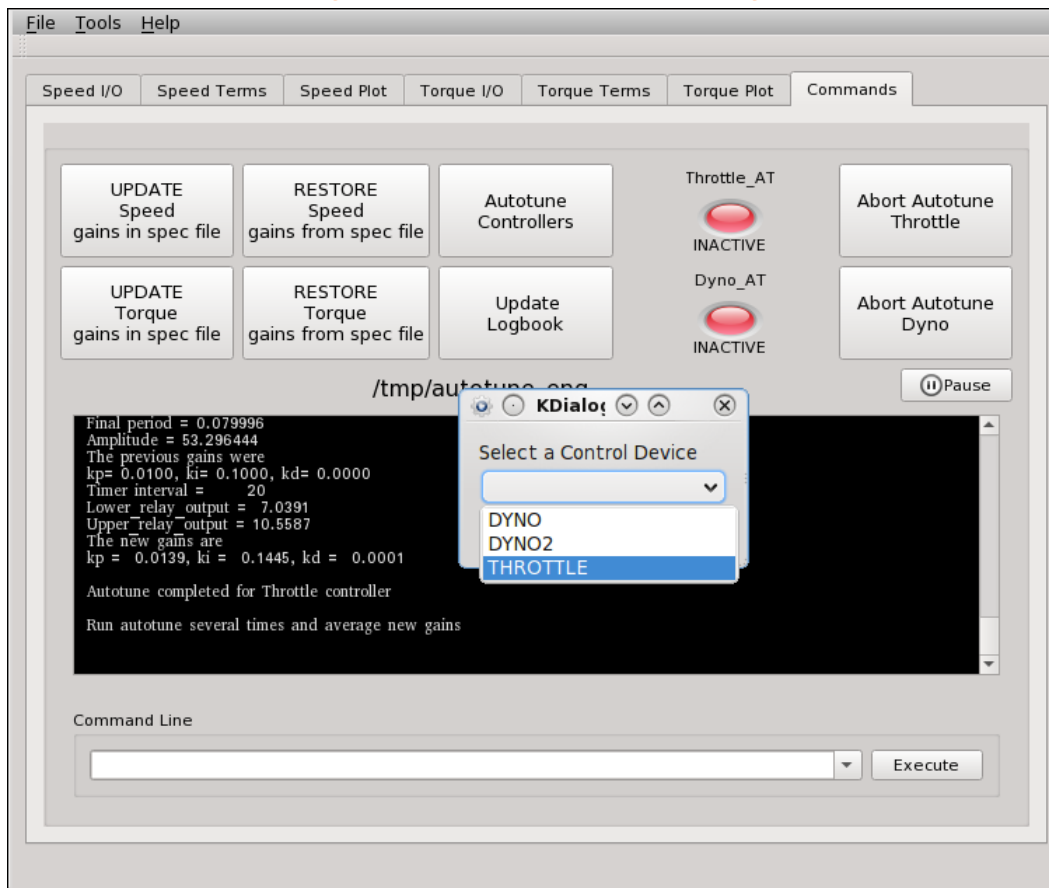


2. Select **Autotune Controllers** from the control display to view the screen shown in *Figure 1*.
3. The program will display the **Select a Control Device dialog** as in *Figure 2*.
  - a. Choose which of the controllers to tune, either **DYNO**, **DYNO2**, or **THROTTLE**. This choice is not necessary for user loops.

**Note:**

Ensure the steady state output is in an appropriate condition prior to the making the selection. The controller will need to change the output by +/- 20% without reaching the output bounds.

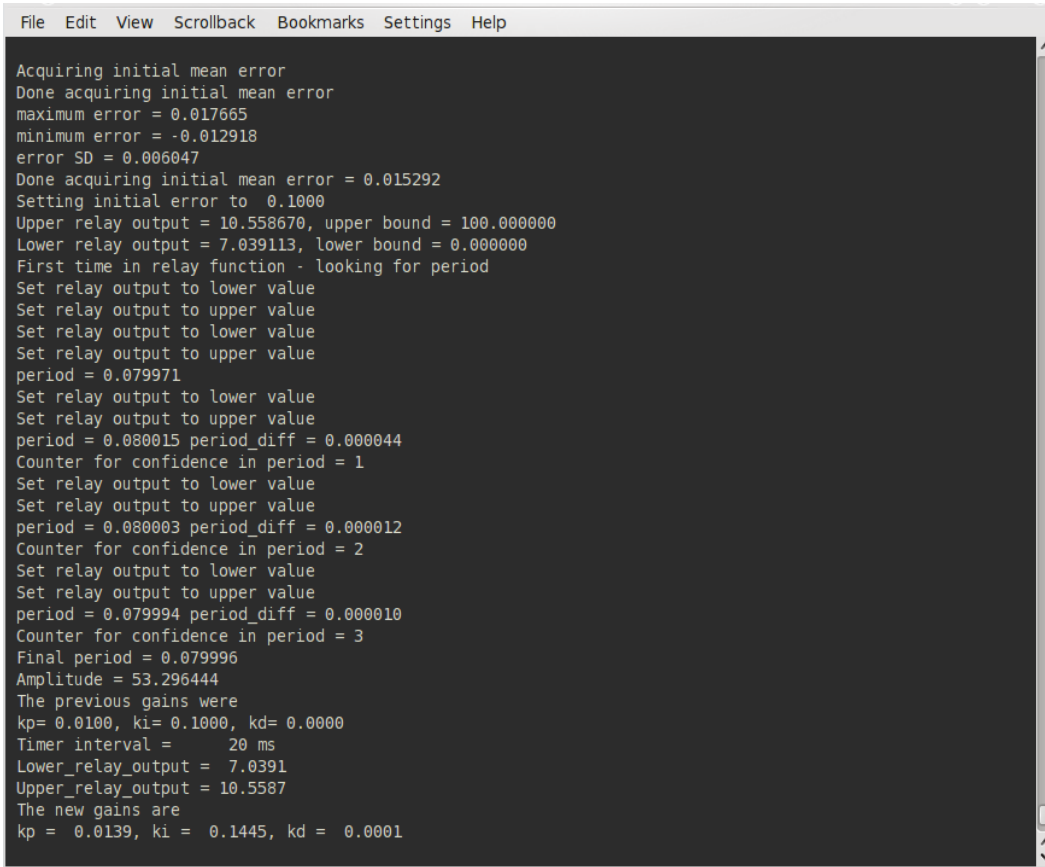
**Figure 2: Select a Control Device Dialog**



4. Select **OK** to start autotuning.

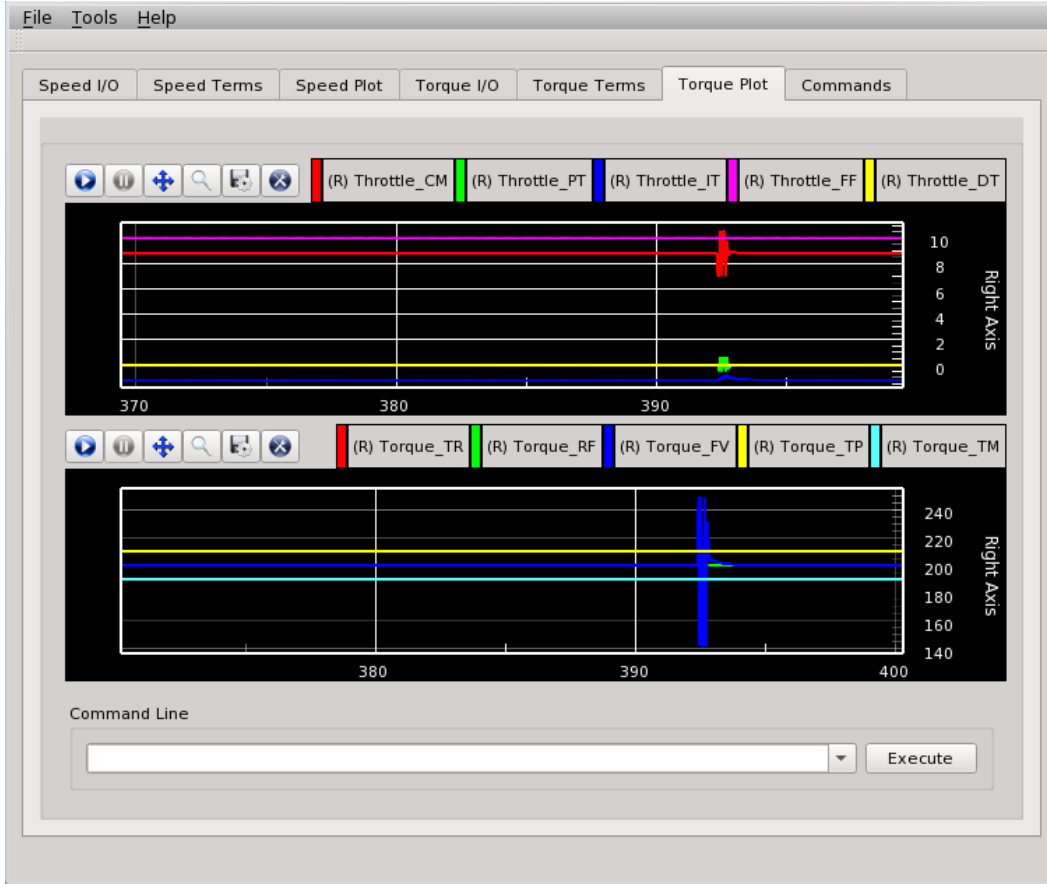
5. During the process:
  - a. As appropriate, select **Abort Autotune Throttle** or **Abort Autotune Dyno** to abort the autotuning process.
  - b. View progress in either the *file tail* screen or the user console. Refer to *Figure 3* and *Figure 4* on page 6 respectively.

**Figure 3: File Tail Progress Display**



```
File Edit View Scrollback Bookmarks Settings Help
Acquiring initial mean error
Done acquiring initial mean error
maximum error = 0.017665
minimum error = -0.012918
error SD = 0.006047
Done acquiring initial mean error = 0.015292
Setting initial error to 0.1000
Upper relay output = 10.558670, upper bound = 100.000000
Lower relay output = 7.039113, lower bound = 0.000000
First time in relay function - looking for period
Set relay output to lower value
Set relay output to upper value
Set relay output to lower value
Set relay output to upper value
period = 0.079971
Set relay output to lower value
Set relay output to upper value
period = 0.080015 period_diff = 0.000044
Counter for confidence in period = 1
Set relay output to lower value
Set relay output to upper value
period = 0.080003 period_diff = 0.000012
Counter for confidence in period = 2
Set relay output to lower value
Set relay output to upper value
period = 0.079994 period_diff = 0.000010
Counter for confidence in period = 3
Final period = 0.079996
Amplitude = 53.296444
The previous gains were
kp= 0.0100, ki= 0.1000, kd= 0.0000
Timer interval = 20 ms
Lower_relay_output = 7.0391
Upper_relay_output = 10.5587
The new gains are
kp = 0.0139, ki = 0.1445, kd = 0.0001
```

Figure 4: User Console Progress Display



**Note:**

On the preceding plot the Dyno command toggles between ~22% and ~30% while the engine speed bounces between 1497.5 rpm and 1502.5 rpm. Because the speed control loop is so responsive in this example, the autotuning is completed in just a few seconds

### 3.1 Using the Results

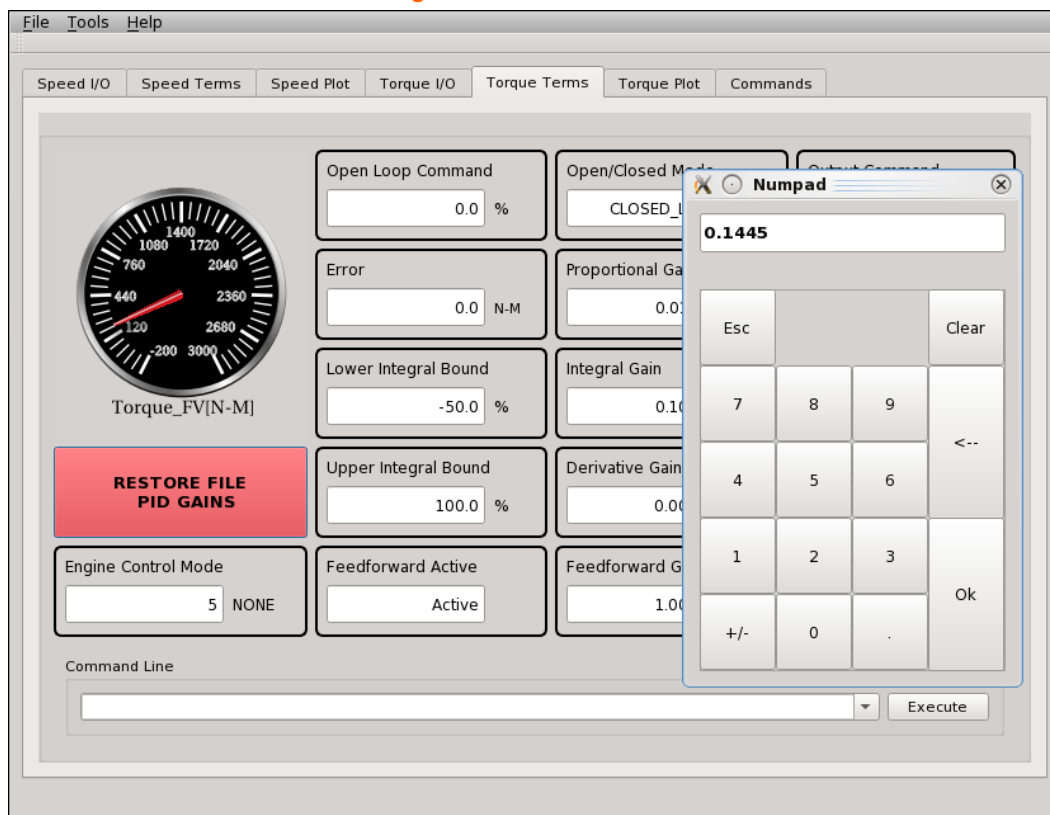
The output of the autotuning process is a new set of P, I, and D gains that will work well for the control loop at the test conditions when started. Before changing the gains, run autotune several times and at different test conditions. Generally, lower suggested gains are the safest to use. Higher gains will make the system respond faster but may cause some overshoot. There is no need to run the old `calc_gains` program.

If the suggested gains vary widely over the entire operating envelop, the lowest suggested gains should be used. If better response is desired, the gains can be modified on the fly with the use of a gain scale variable. Refer to the control system details in *Section 1.1 Background* on page 1 for further information.

Execute the following steps:

1. Select the **Terms** tab for the appropriate controller to change the gains as in *Figure 5*. Double click on the gain the values to use the **Numpad** to enter the values for the P, I, and D gains and try them over a wide variety of operating conditions to make sure that the behavior of the loop is as desired.

*Figure 5: PID Gain Values*



- As preferred, also use the `dy_gains`, `th_gains`, and `u_gains` commands to change the gains to new values.
- Restore the original gains from the specification file at any time by selecting the **RESTORE FILE PID GAINS** button. This is especially useful if an incorrectly placed decimal point sets the gain to a value that causes unstable performance.

2. When satisfied with the response of the system, update the CyFlex Logbook and the specification file. Select the **Commands** tab and then select:
  - **UPDATE Speed/Torque gains in spec file**
    - Enter a WWID in the **E-Logbook Input** dialog as in *Figure 6*.
    - Choose any control specification to update with the **Select Specification file** dialog as in *Figure 7* on page 9. This does not have to be the original file which will be highlighted as in *Figure 7*.

*Figure 6: Update Logbook*

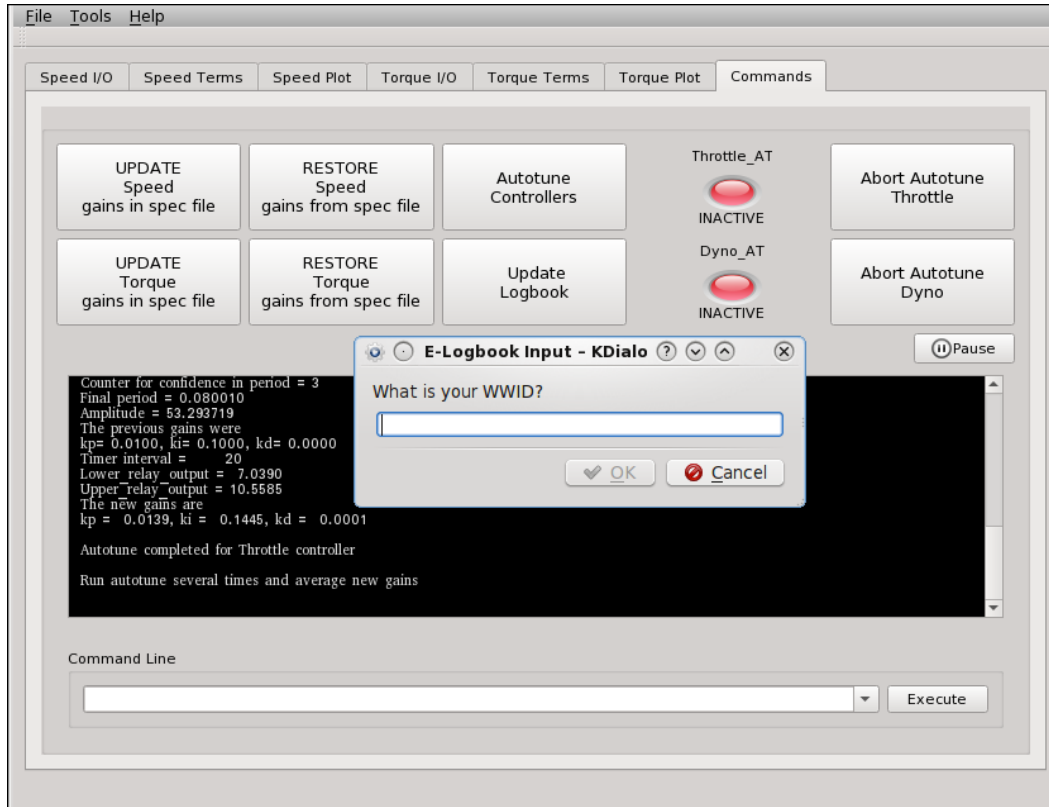


Figure 7: Update Specification File

