

# ServoTune3

## Software for DC servo drive parameters tuning

Software ServoTune3 (Figure 1) is used for parameters tuning for following drives:

- DC servo drive DCS-3010, and
- DC servo drive with analogue output DCS-100-A (use ServoTune3 Ver.3.09 or later version).

Latest version of ServoTune3 software could be downloaded from sites: [www.prizma.rs](http://www.prizma.rs) or [www.audiohms.com](http://www.audiohms.com)

Software consists of one executable file (ServoTune3.exe) and for software installation it is needed to copy this file to a desired folder on PC hard drive.

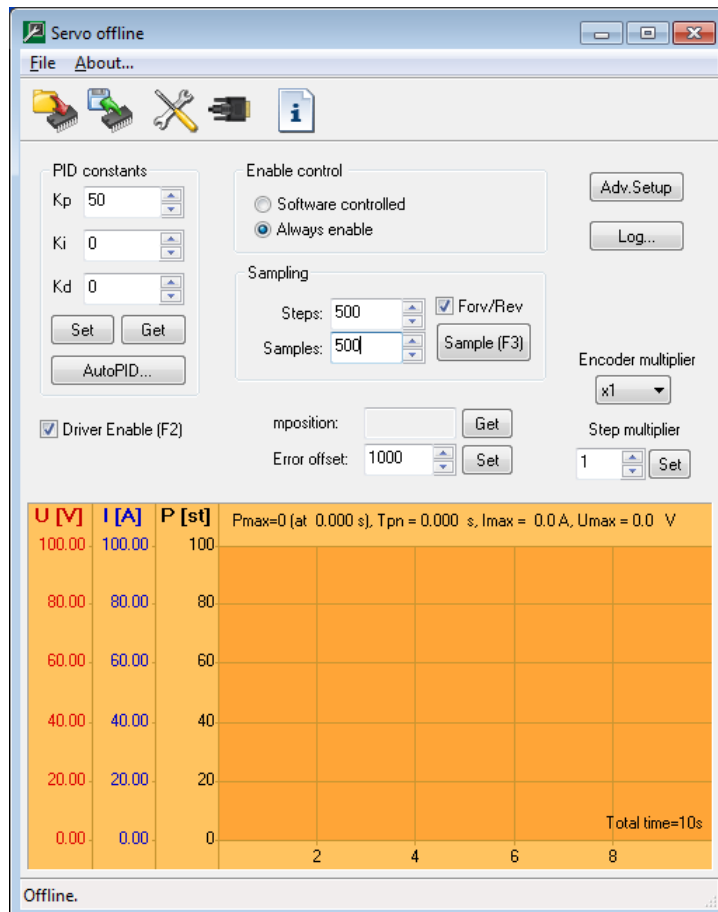



Figure 1 Software ServoTune3

Configuration software  ServoTune3 operates in Windows XP, Windows Vista or Windows 7/8/10 operating systems.

Software ServoTune3 offers options to:

- Adjust constants for PID regulator,
- Adjust resolution multiplier for incremental encoder,
- Adjust step multiplier,
- Enable/disable drive,
- Specify number of steps for sampling of DC motor response to step function,
  - See graphical representation of motor position during step response,
  - See voltage change diagram (not for DCS-100-A), and
  - See current change diagram (not for DCS-100-A),
- Adjust Tracking error offset value,

- Read current position of DC servo motor,
- Record log file containing data for commanded positions, current errors and values of electric current through DC motor (not for DCS-100-A),
- Set type of input interface (STEP/DIR/ENA, CW/CCW/ENA, encoder 1x, 2x or 4x or analog input with or without feedback),
- Choose PWM frequency,
- Adjust digital filter settings for incremental encoder input,
- Set maximum electric current (current limit) for DC motor and more.

**NOTE: ServoTune3 software is used for setting up various working parameters for drive. This software is not intended for controlling DC motor.**

## 1. COM port selection

Connection between software ServoTune3 and PC with drive is achieved via programming interface IPI-USB or PI-BT. Setting parameters for the COM port is done through dialog shown in the Figure 2 which is



accessed by selecting option **File -> Communication setup** or by clicking the icon

Set the number of the COM port where interface for programming is connected as well as the desired baud rate. If check box 'Save to EEPROM' is activated, selected value of baud rate will be recorded in microcontroller's EEPROM.

**NOTE: Uncheck box Bluetooth port if programming interface IPI-USB is used.**

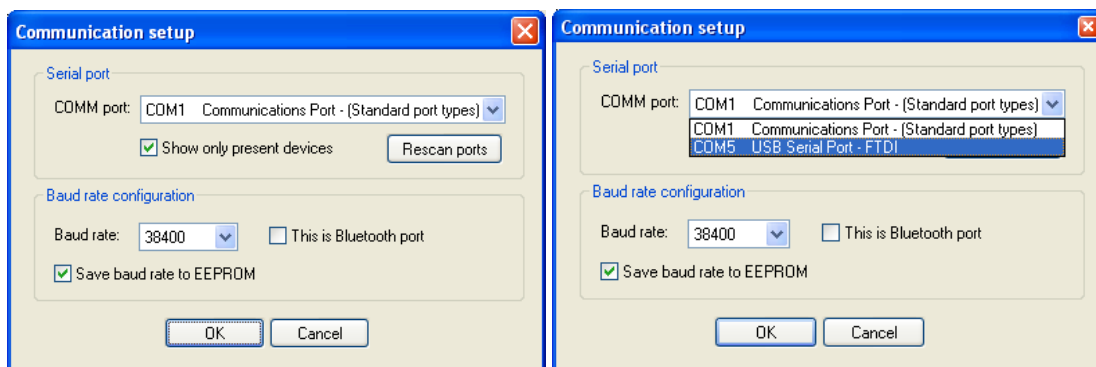


Figure 2 Communication setup dialog

If communication between PC and drive is established, title bar at the ServoTune3 application top will show information that the drive is online with version of firmware (Figure 3). From drive all parameters will be read and shown in the appropriate fields.



Figure 3

In the case that communication is not achieved with the drive, after starting software ServoTune3 warning dialog will appear, as it is shown in the Figure 4 and **Servo offline** status in main window title, as it is shown in the Figure 1.

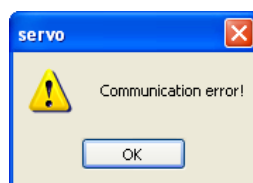


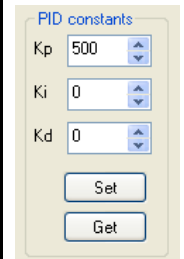
Figure 4 Error opening COM port

This error occurs when the parameters are not set up correctly (number of COM port and baud rate) or when the programming interface is not connected to the drive.

In some cases, loss of communication may occur and then, it is necessary to shut down the ServoTune3 software, restart the drive by pushing RESET button (or power off drive power supply) and start the ServoTune3 software again.

## 2. Setting the PID controller constants

Drives are based on the 16-bit RISC microcontroller with PID control algorithm in it. Setting these constants is performed using fields that are shown in the table below.

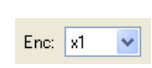
		Name	Description	Minimum	Maximum	Default
Kp	500	Kp	Proportional gain constant	0	32768	50
Ki	0	Ki	Integral gain constant	0	32768	0
Kd	0	Kd	Derivative gain constant	0	32768	0

By pressing the button **Get**, values for Kp, Ki and Kd will be read from EEPROM. To write new values in EEPROM of microcontroller, it is necessary to type desired values of PID constants in appropriate fields and to press the **Set** button.

**NOTE: During setting PID constants take all precautions as it may cause oscillation in DC servo motor – machine system.**

## 3. Setting the encoder resolution multiplication

Drive has the capability for software adjustment of encoder resolution multiplication. So, it is possible to obtain 2 or 4 times higher resolution with incremental encoders with relatively small number of pulses per revolution.

	Name	Description	Values
Enc: x1	Enc	Encoder resolution multiplication	x1, x2 and x4


### EXAMPLE:

Encoder with resolution 500PPR (pulses per revolution) will have:

- 500PPR for encoder multiplication 1x,
- 500PPR x 2 = 1000PPR for encoder resolution multiplication 2x and
- 500PPR x 4 = 2000PPR for encoder resolution multiplication 4x.

## 4. Setting the steps multiplication

Steps multiplier shows how many steps the DC servo motor does for each pulse on the STEP command line. This parameter is useful in case of using the high-resolution encoder, but STEP command generator has no possibility of generating pulses of sufficiently high frequency.

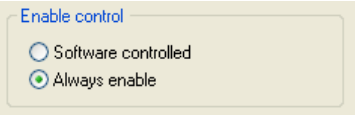
	Name	Description	Minimum	Maximum	Default
1	Step multiplier	Step multiplication	1	50	1

To write desired value in EEPROM microcontroller, it is necessary to press the **Set** button.

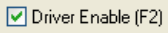
**NOTE:** Higher values for the step multiplier can lead to unsmooth movement, especially at low speed.

## 5. Control of enable input

Options for control of drive enable input are shown in Table below.


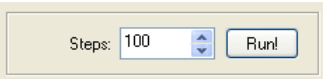
	Name	Alternatives
	Enable control	<p><b>Software controlled</b> – In this mode ENABLE input at the drive control connector Con.1 is activated. If there is logical one (high state) at the ENABLE input, the drive is active and will carry out the commands that come from STEP and DIR command lines. In the case that at ENABLE input is logic zero, then the drive is disabled, commands STEP and DIR are not carried out. This option is often used in some cases if it is necessary to manually rotate the DC motor to desired position.</p> <p><b>Always enable</b> – In this mode ENABLE input at control connector Con.1 is not activated. Drive is always activate (enabled).</p>

From software ServoTune3, selection of ENABLE mode of drive can be performed during parameters setting. The change of check box can be also performed by pressing the function key F2.

	Name	Description	Options
	Drive Enable	Drive Enable	ENABLE – selected DISABLE – not selected

## 6. Record response of DC motor to the step function

In order to adjust the parameters of PID controllers easier, ServoTune3 software provides the ability to record the response of DC servo motor (and attached mechanics to which it is connected) to a step function. In addition, it is possible to set a desired number of steps.

	Name	Description	Parameters		Min	Max
			Name	Description		
	Sampling	Recording response	Steps	Number of steps	1	32767*
			Samples	Number of read values	1	32767
	Steps	Performing a desired number of steps without recording the response	Steps	Number of steps	1	32767*

\* Maximum value of Steps should be lower than set value of Error offset. Otherwise tracking error will appear and DC servo motor will be disabled. Cancelling of tracking error is performed by pressing the RESET button or by switching off the drive power supply.

Executing of appropriate function is performed by pressing the **Sample** or **Run** button.

Pressing the **Sample** button, or by pressing the function key **F3**, DC servo drive will perform the given number of **Steps**. By setting the check box **For/Rev**, for each pressing of **Sample** button DC motor will execute a given number of steps alternating between forward and reverse direction of rotation.

After the execution of Sample command, diagrams of DC motor response to step function will be drawn, and charts of the changes of voltage and current through the DC servo motor will be shown (Figure 5).

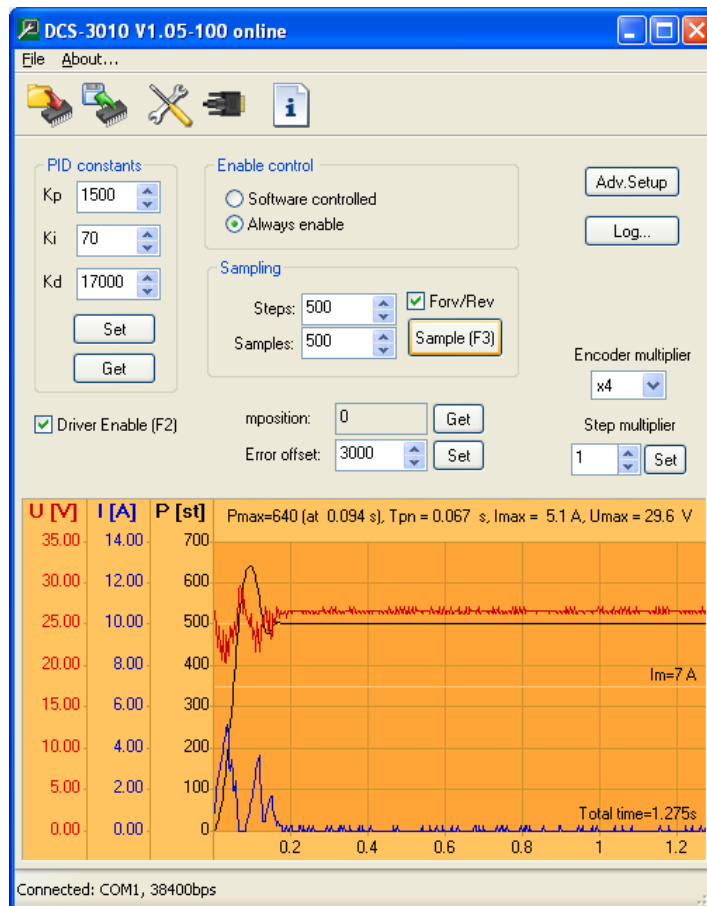


Figure 5

The recorded values of the DC motor position, power supply voltage and current through the DC servo motor will be saved in a file called `odziv.dat` which is located in the folder that contains software ServoTune3. The following is a small sample of the file `odziv.dat`.

```
% ***** ServoTune sampling output *****
% Date and time: 07.12.2012 07:27:39
% Time[s]   Position Current[mA] Voltage[V]
0.000000   0         244      26.63
0.001500   1         488      27.12
0.004000   9         1220     26.13
0.006500   26        1464     24.65
0.009000   48        1953     25.15
0.011500   76        2441     22.19
0.014000   111       2685     24.65
0.016500   152       2685     23.67
0.019000   200       3173     22.68
0.021500   254       3417     20.71
0.024500   315       3906     24.16
0.027000   396       4150     22.68
0.029500   471       4394     20.71
0.032000   553       4638     20.21
```

The first column of the file is the time, the second column is the current position of the DC motor, the third column represents the value of the current through the DC motor in milliamps (mA) and the fourth column is the change of supply voltage of DC motor in volts (V). The values from the file `odziv.dat` can easily be imported into the software for drawing diagrams (Excel, MATLAB, i.e.).

## 7. Setting values of tracking error offset

Setting of tracking error offset can be performed by entering the desired value in a field named **Error offset**.

Error offset: <input type="text" value="1000"/> <input type="button" value="Set"/>	Name	Description	Minimum	Maximum	Recommendation
	Error offset	Tracking Error offset	0	32767	bigger than 100

To save desired value of tracking error offset into EEPROM it is necessary to press the **Set** button.

If difference between controlled position and the current position of DC servo motor exceeds the set tracking error offset value, Tracking Error output activates (at control port), and **OP/ER** indicator will indicate this error. Additionally, DC servo motor will be DISABLED. Cancelling the error of tracking offset is achieved by pressing RESET button or switching off the drive power supply.


## 8. Read the current value of the DC motor position

Reading current position of DC servo motor, i.e. encoder position (**mposition**) is obtained by pressing the button **Get**.

mposition: <input type="text" value="0"/> <input type="button" value="Get"/>	Name	Description
	mposition	Current position of DC servo motor

## 9. Saving and loading configurations

Once adjusted configurations can be saved in a configuration file by selecting **File -> Save config...** or by

pressing the button .

Also the configuration file with all the settings can be loaded into the drive by choosing option **File -> Load**

**config...** or by pressing button .

## 10. Advanced Setup

By pressing the button **Adv. Setup** (Figure 6) or icon  dialog box is opened which offers various options for advanced settings (Figure 7).

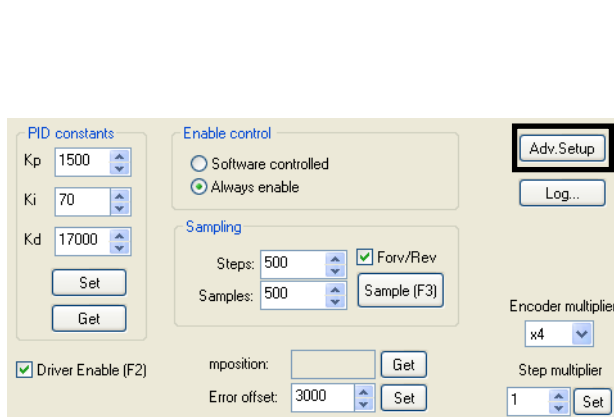


Figure 6

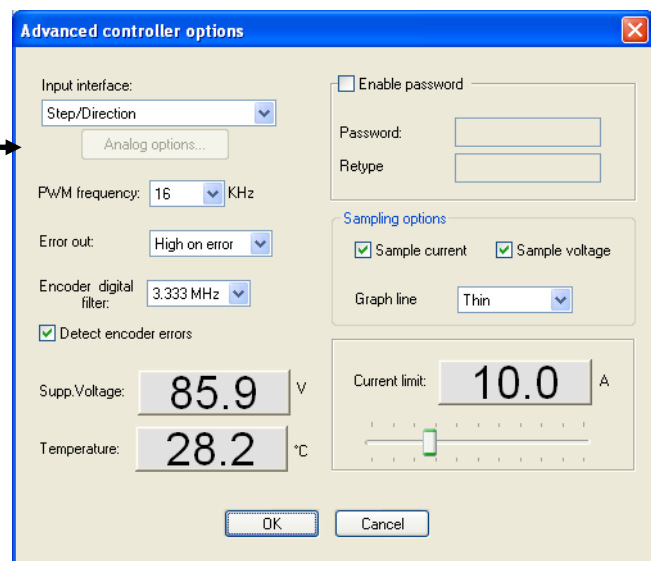


Figure 7

Advanced settings include the range of options:

- Type of input interface,
- PWM frequency,
- The logic level at Error output in case of error,
- Digital filter setup for incremental encoder,
- Incremental encoder error detection option,
- Password protection,
- Selection of parameters that will be shown in the main diagram and
- Maximum current of DC motor – current limit (not for DCS-100-A).

In order to accept change of any of the above values in the EEPROM of microcontroller it is necessary to press the button OK (Figure 7).

## 10.1 Type of Input interface

Control of DC motor is carried out using three input command lines on drive. The first two command lines in this manual are called STEP/DIR, while the third is ENABLE. Option to choose type of input interface provides a choice of control modes via the above three command lines, or via the analogue inputs in the following modes:

- Step/Direction i.e. STEP/DIR/ENABLE,
- StepUp/StepDown i.e. CW/CCW/ENABLE,
- Encoder x1 /ENABLE,
- Encoder x2 /ENABLE,
- Encoder x4 /ENABLE,
- Analogue input with feedback (Analogue with FB) and
- Analogue input without feedback (Analogue without FB).

**NOTE: Configuration of input interface type Encoder 1x, 2x and 4x are not fully tested.**

Drive has the ability to control DC motor by means of voltage signal of 0+5V which is applied to the analogue input (see drive user manual). At the analogue input it is possible to directly connect potentiometer or external motion path generator (see drive user manual).

If you choose one of the options that uses analogue input, **Analogue option** button will become active, and after activating it one of dialog boxes (shown in Figure 8) will appear.

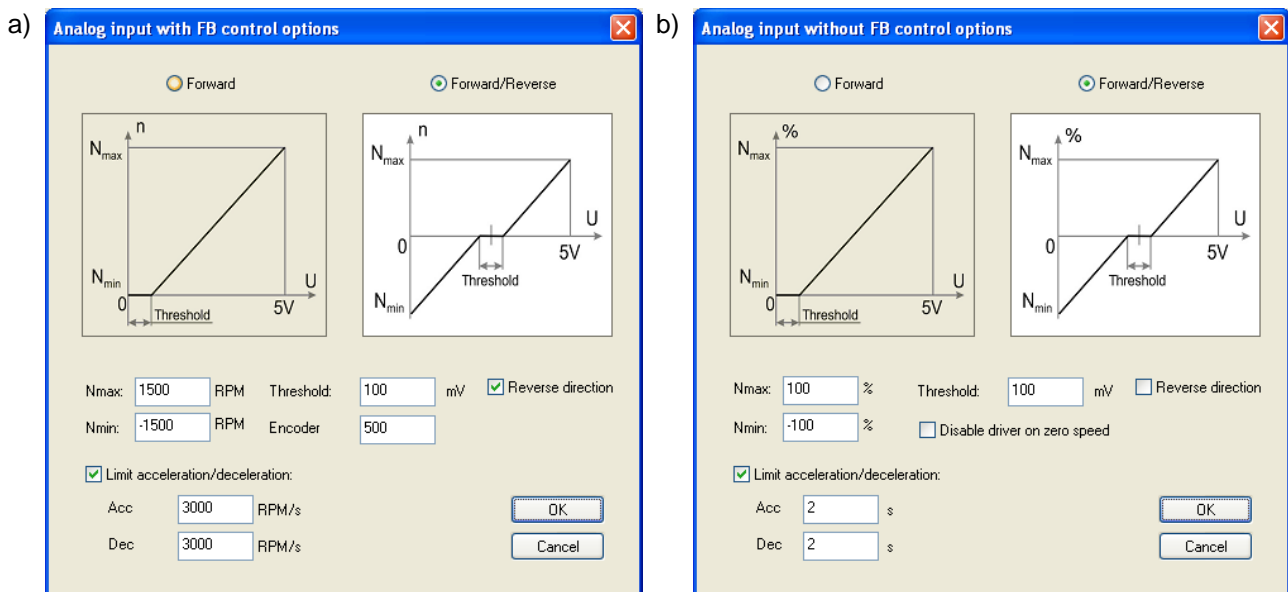


Figure 8 Dialog box for setting parameters, a) analogue input with feedback (Analogue with FB) and b) analogue input without feedback (Analogue without FB)

Dialog box for setting analogue input parameters provides a choice:

- One or both directions of DC motor rotation (Forward or Forward/Reverse),
- Change the direction of DC motor rotation (Reverse direction),
- Maximum (**Nmax**) and minimum (**Nmin**) revolution speed.
  - In case of choosing the analogue input with feedback (Analogue with FB) values **Nmax** and **Nmin** are in revolutions per minute (RPM).
  - In case of choosing the analogue input without feedback (Analogue without FB) values **Nmax** and **Nmin** are in percent (%) in relation to power supply voltage of DC motor.
- Width of dead (inactive) zone (**Threshold**) expressed in mV.
- Incremental encoder lines count (**Encoder**) in case of choosing analogue input with feedback (Analogue with FB).
- Option to disable DC motor in case that adjusted speed is equal to zero in case of choosing analogue input without feedback (Analogue without FB).
- Parameters for acceleration (**Acc**) and slowing down (**Dec**) (Limit acceleration/deceleration).
  - In case of choosing the analogue input with feedback (Analogue with FB) values **Acc** and **Dec** are in revolutions per minute per one second (RPM/s).
  - In case of choosing the analogue input without feedback (Analogue without FB) values **Acc** and **Dec** are in second (s).

## 10.2 Setting the PWM frequency

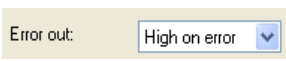
This option provides the ability to adjust the PWM frequency:

- 10 kHz,
- 12 kHz,
- 14 kHz,
- 16 kHz (default),
- 18 kHz,
- 20 kHz, and
- 32 kHz (Only available for DCS-100-A. Recommended value for DCS-100-A)

For PWM frequencies bellow 20 kHz audible "whistling" can be heard coming from DC motor (not for DCS-100-A).

## 10.3 Logic level on Error output in case of error

The selection of logic level in case of errors at Error output is presented in the table below.

	Name	Option of choice – description
	Error out	<p><b>High on error</b> – In the case of error – Error output will be at logical high level</p> <p><b>Low on error</b> – In the case of error – Error output will be at logical low level</p> <p><b>Always low</b> – Error output will always be at logical low level independently of the presence of errors</p>

## 10.4 Digital filter for incremental encoder

An adjustment of digital low pass filter for incremental encoder input is performed through selection of upper cut-off frequency:

- digital filter switched off (Turn OFF),
- the frequency of filter 6.667 MHz,
- the frequency of filter 3.333 MHz (default),
- the frequency of filter 1.667 MHz,
- the frequency of filter 416.7 kHz,
- the frequency of filter 208.3 kHz,
- the frequency of filter 104.2 kHz,
- the frequency of filter 52.1 kHz i
- the frequency of filter 26.0 kHz.



Option for digital filtering of signals from the incremental encoder can be useful in the environment with strong electromagnetic interference, which can lead to errors in reading the incremental encoder position.

### 10.5 Detecting encoder errors

If this option is enabled, the drive checks if there is a change of signal level on both encoder inputs (A and B). If this is not the case (e.g. only one input changes and the other remains constant), the drive output stage will be disabled and OP/ER LED indicator will show the encoder error.

**NOTE: This option has not been fully tested, and it is recommended that the check box remains off.**

### 10.6 Reading power supply voltage of DC motor and temperature of drive

In the appropriate fields (fields in the Figure 9 with rectangular marks) it is possible to read power supply voltage of DC motor in volts (Supp. Voltage) and temperature of servo drive close to the microcontroller in °C (Temperature).

**NOTE: This function is not available for drive DCS-100-A.**

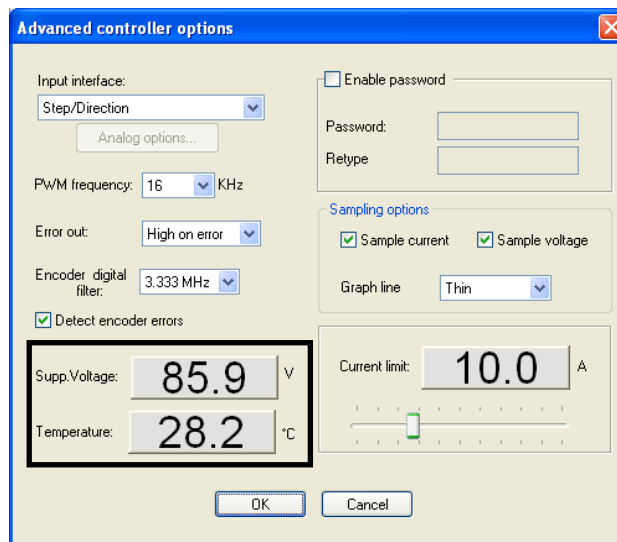


Figure 9

### 10.7 Entering the password

The ServoTune3 software provides the ability to enter password to prevent unauthorized changes of drive parameters.

In order to activate this option it is necessary to select a check box **Enable password** first as it is shown in the Figure 10 thereby **Password** and **Retype** become active and in this fields it is then possible to enter the password. When the password is entered then during each of the next start-up of ServoTune3 software and the connection to the drive, dialog window (Figure 11) for the entering password will open, which becomes a requirement to allow settings.

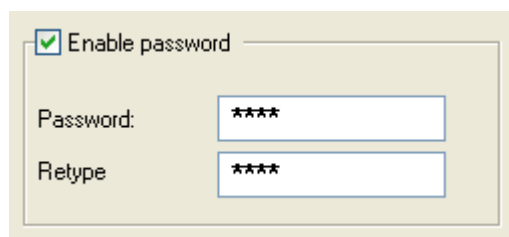


Figure 10 Password option group



Figure 11 Password dialog box on ServoTune3 software

## 10.8 Selection of options of the main diagram

The part of dialog in the advanced settings (Advanced setup) shown in the Figure 12 (Sampling options) allow choosing of:

- What parameter will be shown in the main diagram and
- The thickness of the diagram line that will be plotted (Thin, Medium and Thick).

**NOTE: DC motor current and DC motor voltage cannot be reading from drive DCS-100-A.**

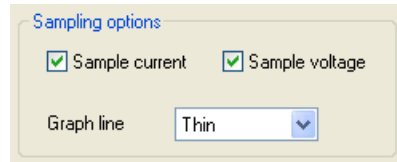


Figure 12

## 10.9 Setting the maximum DC motor current

Using the slider shown in the Figure 13, maximum current of DC motor can be set. The adjusted value is readable in the appropriate field.

Maximum current through DC motor can be adjusted in the range of 3÷30 A (for DCS-3010).

**NOTE: This function is not available for drive DCS-100-A.**

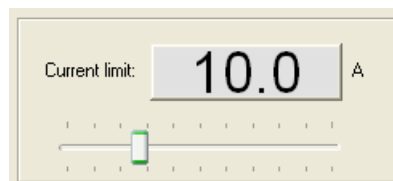


Figure 13

## 10.10 LOG File recording

Recording of Log file is activated by pressing the **Log** button (Figure 14). This opens the **Logging** dialog (Figure 15).

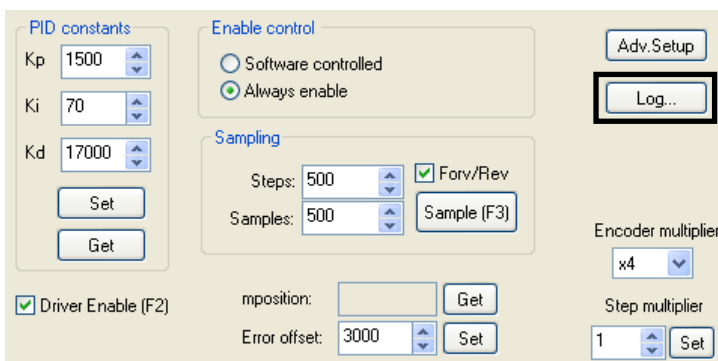


Figure 14 Position of **Log** button

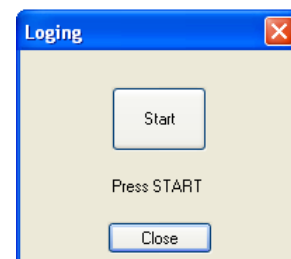


Figure 15 Log dialog

Start recording a log file is activated by pressing **Start**, and **Stop** recording by pressing this button again. Data will be stored in a file called `servo.log` that is located in the same folder as the ServoTune3 software. The following is a small part of the `servo.log` file.

```
% ***** ServoTune log output file *****
% Date and time: 07.12.2015 19:00:05
% Time[s]    Position  PosDiff Current[mA] Voltage[V]
0.020960    -401      0         0         98.23
0.023580    -401      0         0         98.23
0.025676    -401      0         0         98.23
0.027772    -401      0         0         97.78
0.029868    -401      0         0         98.23
0.032488    -401      0         0         98.23
0.034584    -401      0         0         98.89
0.036680    -400      1         0         98.23
0.000000    -396      5         0         98.23
0.001572    -386     13         0         98.23
0.003668    -373     20         0         97.78
0.005764    -349     32        119        97.02
0.007860    -325     39        833        95.45
0.009956    -296     44       1310        94.02
0.012576    -263     39       1905        93.25
0.014672    -226     38       2381        92.15
```

At that:

- **The first** column is time in seconds,
- **The second** column is present position of DC motor,
- **The third** column is position error (tracking error), i.e. difference between specified and current position of DC servo motor. It is expressed in the steps,
- **The fourth** column represents the value of the current through the DC motor expressed in mA (not for DCS-100-A), and
- **The fifth** column is supply voltage of DC motor expressed in V (not for DCS-100-A).

Data from file `servo.log` file can be easily loaded into any software for drawing for further analyzing. (Excel, MATLAB etc.).

In the Figure 16, Figure 17 and Figure 18 it is shown an example of parameter changes from one recording `servo.log` data file.

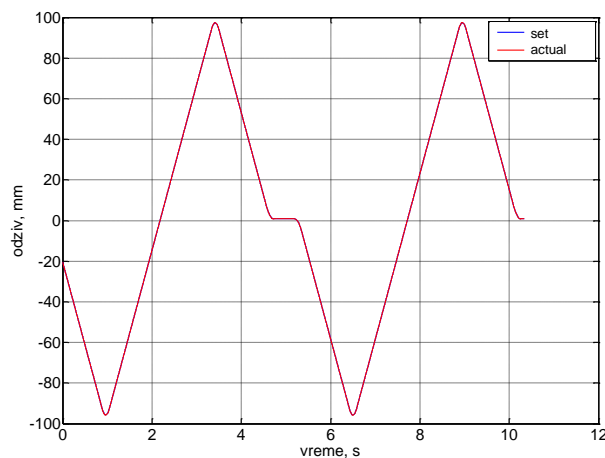


Figure 16 Diagram of set and actual position of DC servo motor calculated in mm

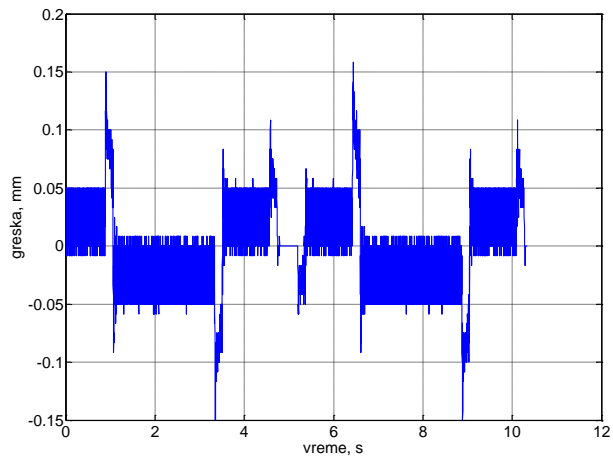


Figure 17 Diagram of the positioning error calculated in mm

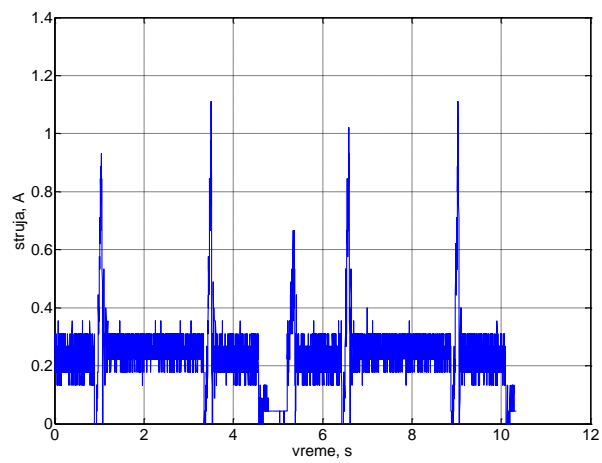


Figure 18 Diagram of the electric current through the DC servo motor (not for DCS-100-A)

## 11. PID CONSTANTS SETUP PROCEDURE

**NOTE: During adjustments of PID constants take all precautions as it may cause oscillation of DC servo motor-mechanics.**

**When setting up, begin with setting lower value of maximum current through the DC motor and continuously monitor behaviour of DC motor. After this, gradually increase the value of maximum DC motor current (not for DCS-100-A).**

The values of the PID constants depend on:

- Characteristic of DC motor (moment of inertia, supply voltage, maximum electric current, etc.),
- Physical characteristics of mechanics, which is connected at DC motor (mass, damping, etc.),
- Resolution of the incremental encoder mounted on the DC motor,
- Selected incremental encoder resolution multiplication (x1, x2 or x4), and
- Some other factors (ambient temperature, lubricant, etc.).

Adjusted PID constants are valid only for that configuration. If there is a change of configuration (some of the above parameters) it is necessary to readjust the PID constants.

The following is a description of the adjustment of PID constants. Incremental encoder resolution in this case is 500 PPR, encoder resolution multiplication is x4, so that the overall incremental encoder resolution is 2000 PPR.

**STEP 1:** Initial value of PID constants is:

- Proportional constant  $K_p = 50$ ,
- Integral constant  $K_i = 0$
- Derivate constant  $K_d = 0$ .

Slowly increase the constant  $K_p$  until you get response similar to the response shown in Figure 19.

**STEP 2:** Increase constant  $K_d$  until the system response becomes more stable as is shown in Figure 20. The constant  $K_d$  may be significantly larger than the constant  $K_p$ .

**STEP 3:** Gradually increase the constants  $K_p$  and  $K_i$  to the response as shown in Figure 21. Here it should be noted that the constant  $K_i$  is much smaller compared to the other two constants.

**Repeat steps 2 and 3 to the point where the shaft of DC motor behaves as “locked“. Also check the motor running at various revolution speeds. If PID constants are set correctly than oscillation and vibration during operation will not occur.**

**STEP 4:** Final values of the PID constants and the response of the motor are shown in Figure 22.

It is necessary to check determined PID constants during the work of the machine and, if it is necessary, to correct them.

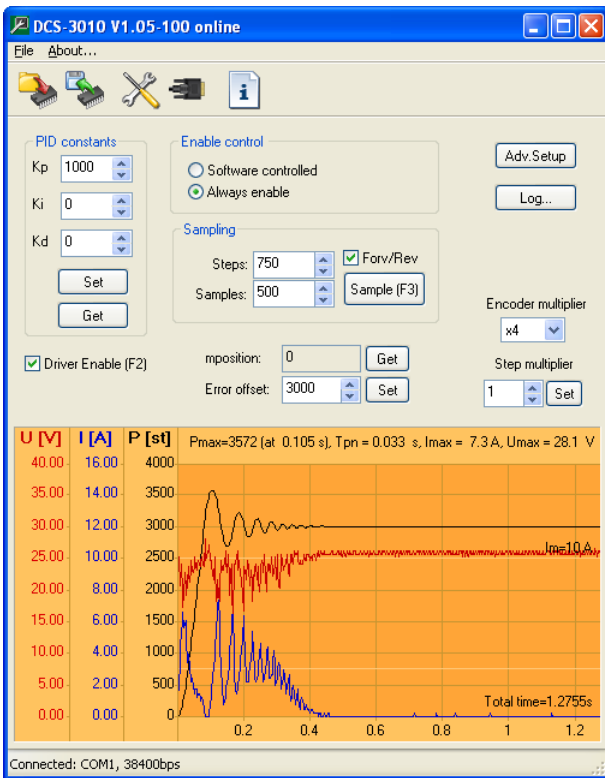


Figure 19

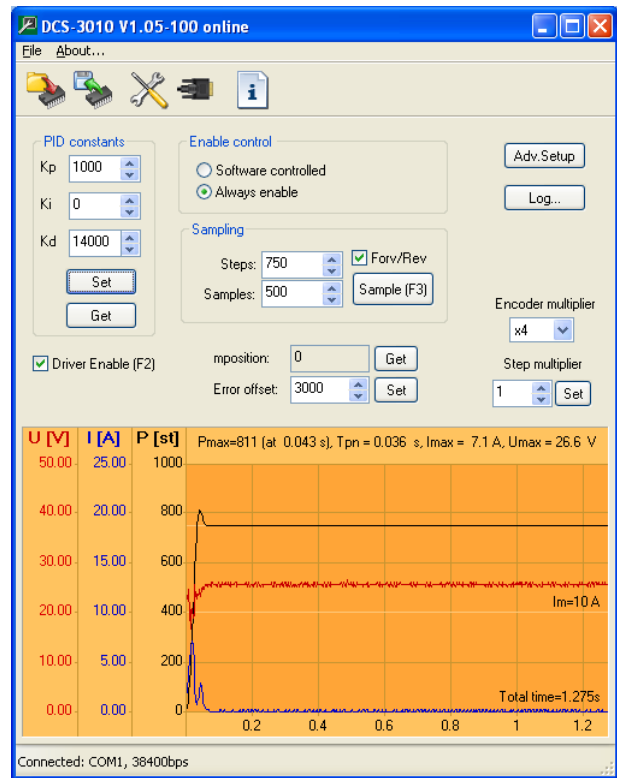


Figure 20

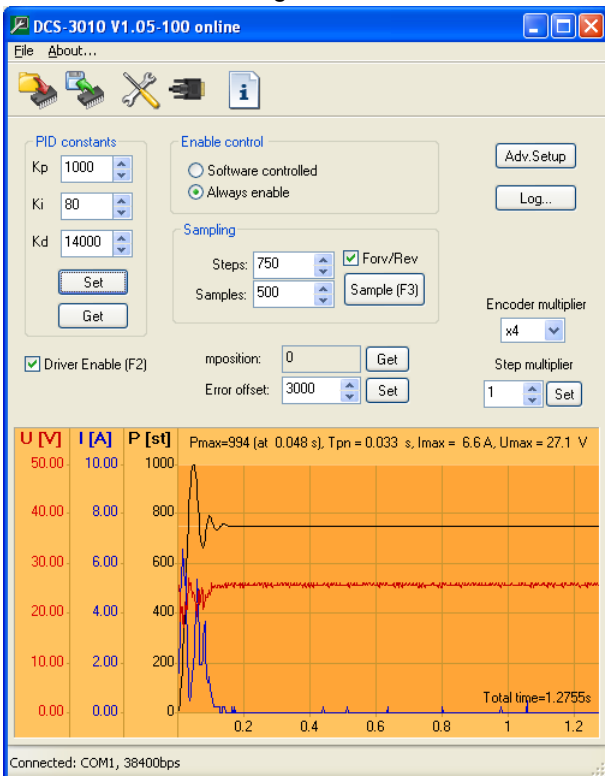


Figure 21

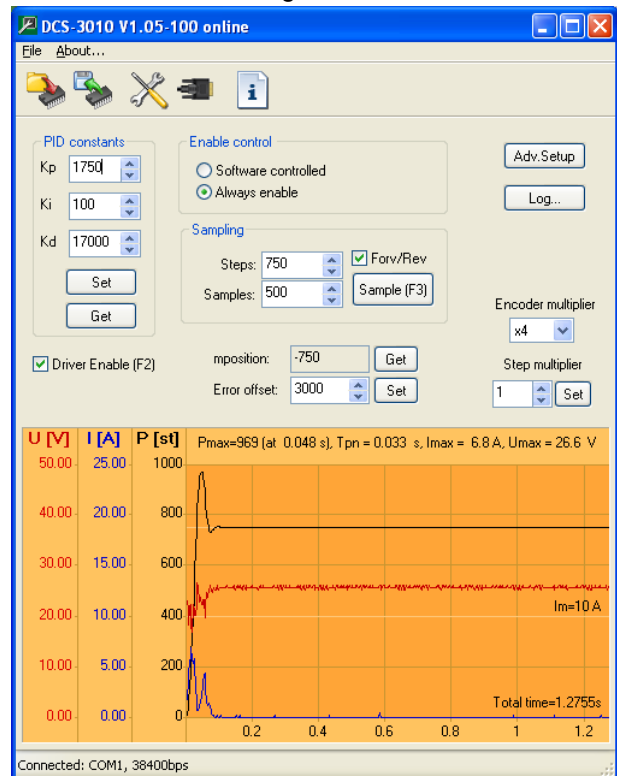


Figure 22

## 11.1 Automatic adjustment of PID constants

The auto-configuring and adjusting of PID constants is available from ServoTune3 software version v3.07 and above. The dialog for automatic adjustment of PID constants opens by pressing button **AutoPID** (Figure 23). This will display a warning dialog shown in the Figure 24.

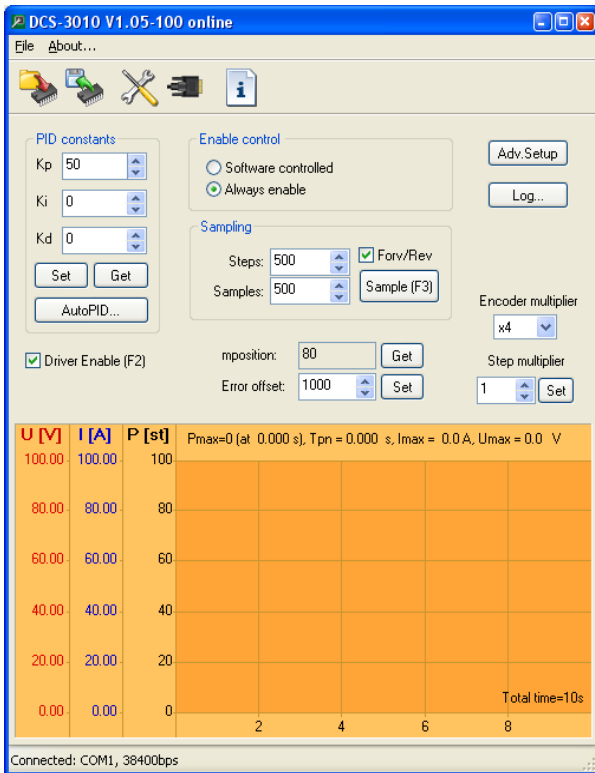


Figure 23

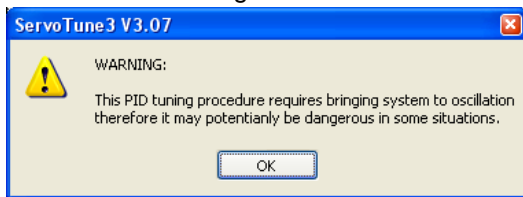


Figure 24

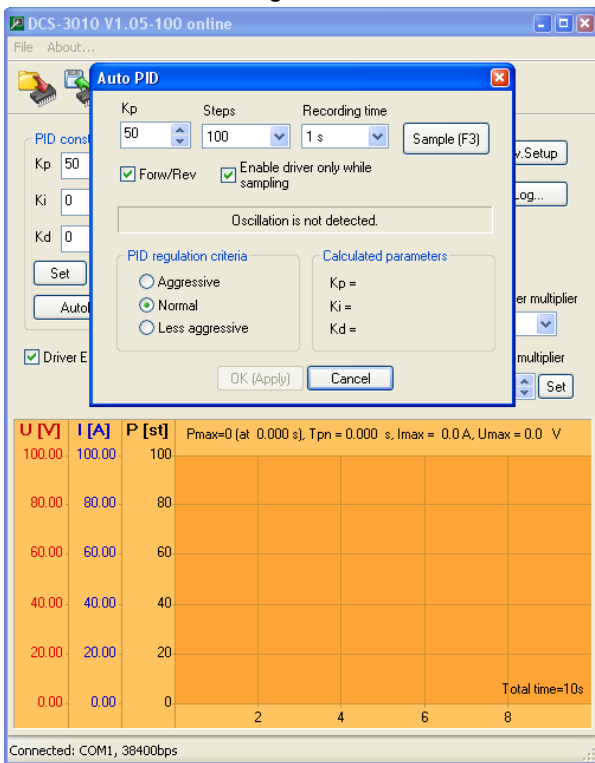


Figure 25

**NOTE: Procedure for automatic adjustment of PID constants involves bringing DC servo drive – DC servo motor system and mechanical equipment in an unstable state, so it is necessary to mention that the system will oscillate. Setup procedure of PID constants in this way is performed by user at his own risk.**

Within **AutoPID** dialog, which is shown in the Figure 25, it is possible to set the following parameters:

- Proportional constant **Kp**.
- Number of **Steps** to define the step function. Pull-down menu provides a choice of predefined values for the number of steps: 100, 150, 200, 250, 300, 400, 500, 750 and 1000. Choose one value of the number of steps corresponding to approximately 5-10% of the number of steps required for DC motor to make a full revolution.
- EXAMPLE: Encoder has 500PPR and selected the option Encoder multiplier x4. In this case it is necessary 500 x 4 = 2000 steps to the DC motor make a full revolution. Recommended values for Steps would be 100, 150 or 200.**
- The duration of the measurement (**Recording time**) that can be selected via the pull-down menu: 1s, 2s, 3s, 4s and 5s. Considering that during the recording of system response it is required to bring the system into oscillation, it is recommended for this time to be as short as possible.
- Alternately reversing DC motor when recording step function (option **Forw/Rev**). It is recommended that this option should be active.
- Option to **Enable drive only while sampling**. It is recommended that this option is activated.

The process of automatic adjustment of the of PID constants involves gradually increasing the constant **Kp**. With every change of constant **Kp**, it is necessary to press the button **Sample (F3)** or function key **F3** to record the system response. The constant **Kp** is increased gradually and gently, until it comes to the appearance of the oscillation of the system, such as that shown in the Figure 26, Figure 27 and Figure 28. It is important to note that in these figures not all the steps are shown for gradual increase of the constant **Kp**.

ServoTune3 software will recognize that there is a system oscillation, as is shown in the Figure 28 (**Oscillation is detected...**) and will suggest PID controller constants **Kp**, **Ki**, and **Kd** according to the given criteria. It is possible to choose one of the following criteria:

- **Aggressive**,
- **Normal** or
- **Less aggressive**.

Pressing the **OK (Apply)**, calculated PID constants will be stored in EEPROM of microcontroller.

Check behaviour of DC servo motor with applied PID constants calculated in this way (Figure 29).

If necessary, perform manual fine tuning of PID constants.

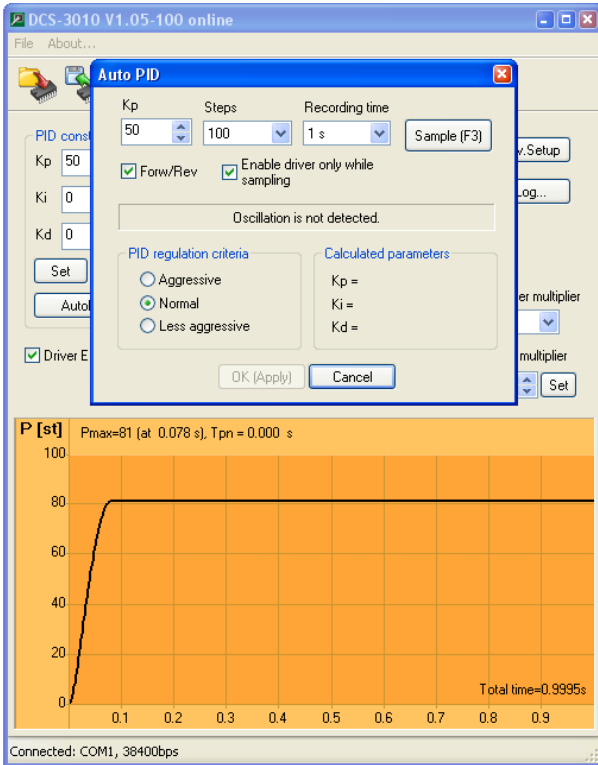


Figure 26

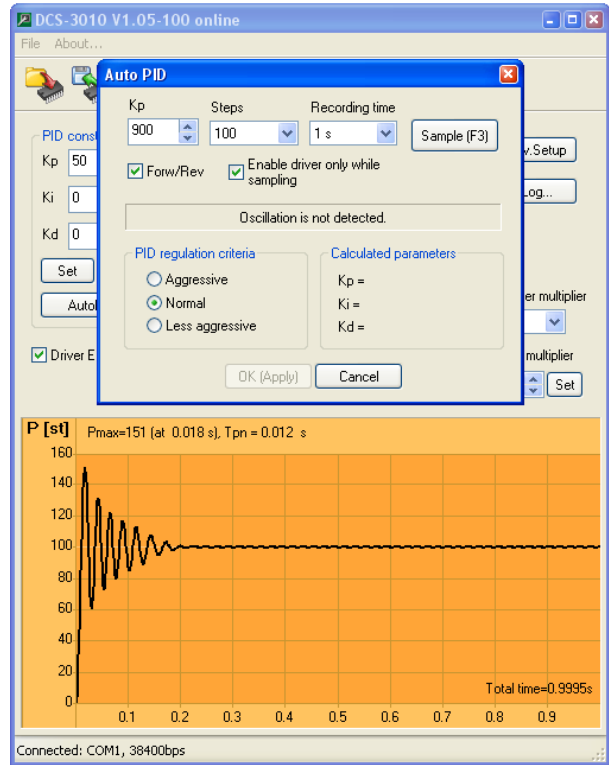


Figure 27

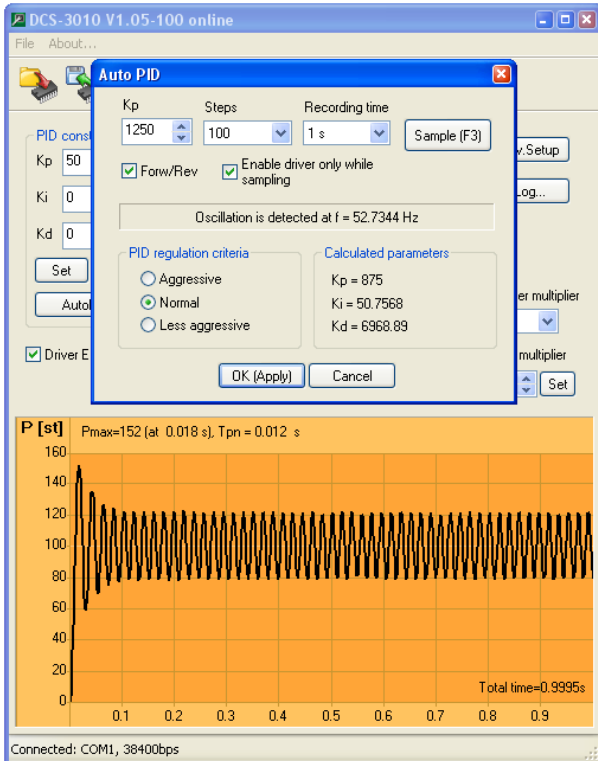


Figure 28

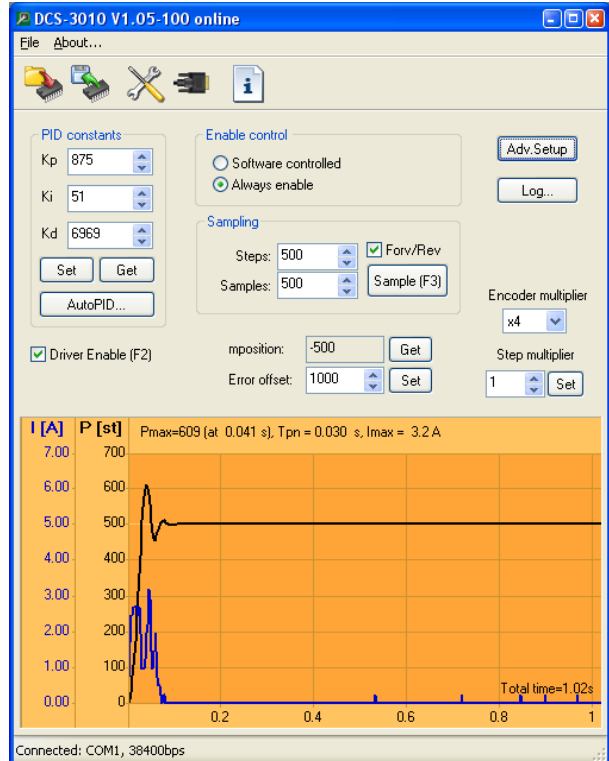


Figure 29

**DOCUMENT REVISION:**

- Ver. 1.0, June 2016., Initial version

