## Temporal Beam Shaping

The temporal shape of the laser pulse can be adjusted through control of the Arbitrary Waveform Generator (AWG) in the front end. A target pulse shape is selected, and a feedback loop adjusts the AWG waveform to drive the amplified pulse to the desired temporal profile. The loop is closed through a measurement of the pulse shape at the output with a photo-detector coupled to an oscilloscope.

### Configuring the feedback loop

Figure 1 shows the user interface for configuring the parameters for the feedback loop.

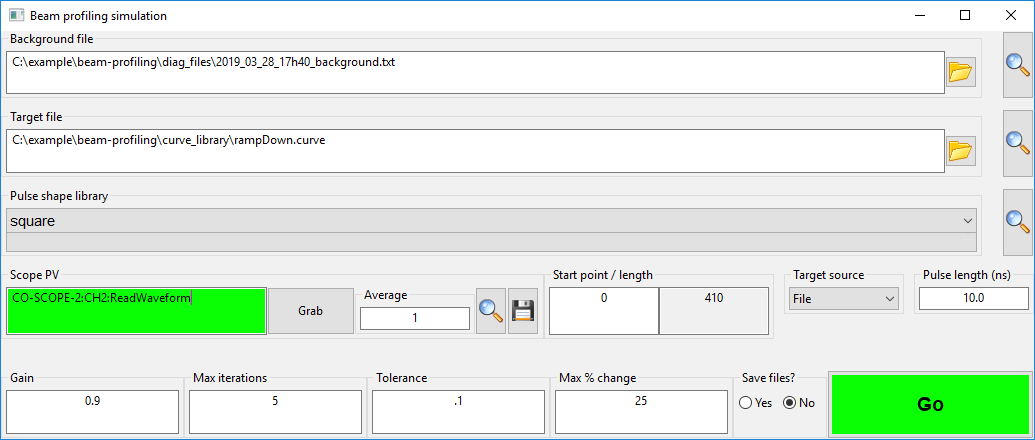


Figure 1: User-interface for configuring the feedback loop

In this window, a background file must be specified (a waveform taken from the feedback oscilloscope in the absence of a laser pulse). It is also necessary to choose a target shape on which the loop will attempt to converge. This can be done either by loading a shape form a file, or by choosing from pre-defined library of pulses. The format of these files, and instructions on adding to the pulse library, are detailed in Section 1.1.6.

The selected background and target curves may be previewed with the magnifying glass button. This will launch a graph window showing the data, or an error message if the file couldn’t be loaded. Only one graph window can be loaded at a time and previewing more than one file will show them on the same set of axes. To clear the graph, close the window before previewing other file.

The pulse shape library is a drop-down list of all pre-defined shapes available. The decision to use either the pulse-shape library or a user-loaded target file is made with the **Target Source** drop-down.

Data for the output pulse is read from the process variable (PV) defined in the **Scope PV** field. If this PV is connected, the background for the field will be green. If not, then it will be purple. This is shown in Figure 2.

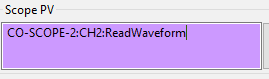
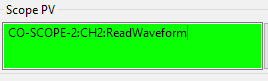
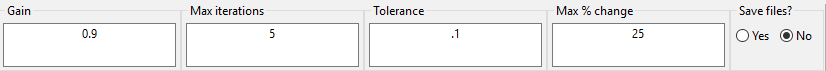


Figure 2: Background colour for a connected (Left) and disconnected (Right) PV

Once the correct PV is connected, data is read from the scope using the **Grab** button. Each grab will be an average over the number of acquisition specified in the **Average** field. The data acquired can be previewed as above, and saved using the **Save** button.

Fundamental to the success of the feedback loop is matching each of the AWG samples with the correct time in the output pulse. This is done using the **Start point/length** control. The start point is the pixel number in the oscilloscope trace that corresponds to the first sample of the AWG. Details on how to determine this are in Section 1.1.7. The required **Pulse Length (ns)** is entered by the user, and the necessary number of data points for the feedback loop is automatically calculated from the oscilloscope time base (read via a PV determined from the **Scope PV** name) and the time resolution of the AWG. This **length** field is read-only, and is shown for information.

The bottom row of specifies the control parameters for the loop. The calculated correction for each loop will be scaled by the value specified in the **Gain** field (max value is 1), subject to the limitation that each point in the AWG waveform may not change by more than **Max % change** in a single step (this restriction does not hold when setting a point to zero). The loop will automatically terminate when either the number of iteration reaches **Max iterations**, or the RMS error between the target and output is less than **Tolerance**. Diagnostic files may be saved for each iteration of the loop, showing the AWG waveform, output data and applied correction for each iteration, along with the target and background data. The option to write these files is controlled with the **Save Files?** radio button.

### Running the loop

Once the parameters have been defined, the loop is launched by clicking on the **Go** button, and a second window opens showing graphs for the loop data. This is shown below in Figure 3.

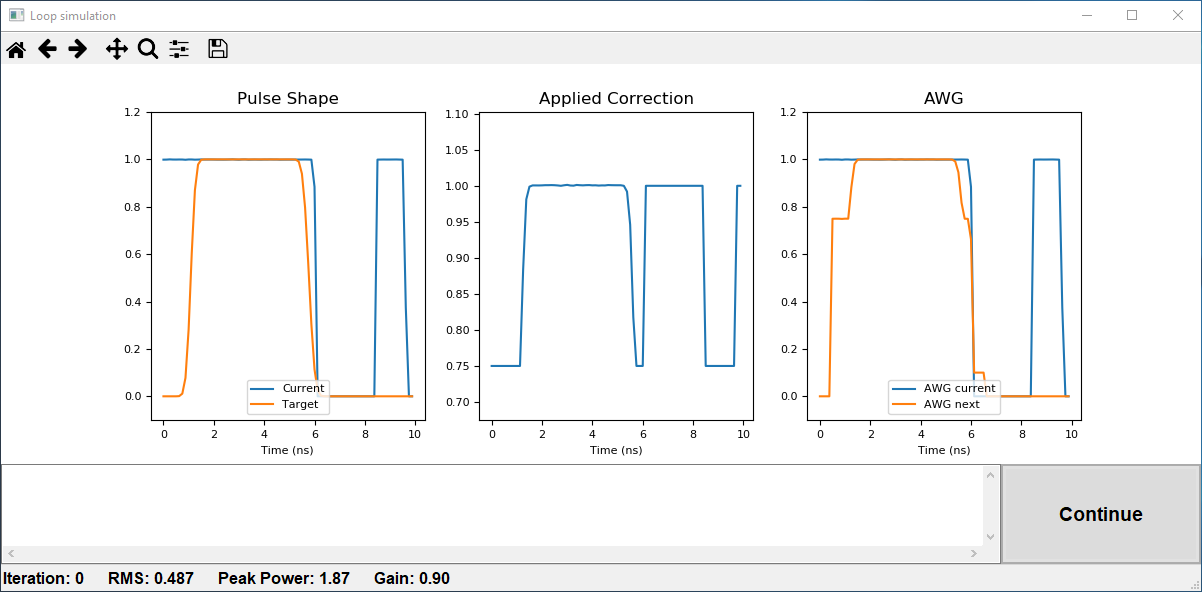
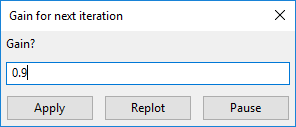


Figure 3: The window that displays while the feedback loop is operational

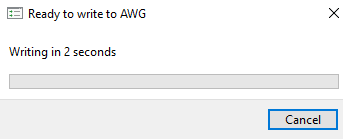
The **Pulse Shape** graph on the left shows the current output, and the desired output (target). The RMS error between the two is calculated and shown in the status bar. A value for the peak power is also calculated and tested against a safe value set by the user. If the safe value is exceeded the loop will terminate. Section 1.1.3 shows where to set this safe value.

The centre graph shows the **Applied Correction**, a multiplication factor that will be applied to each of the points in the AWG waveform in the next iteration. The current output of the AWG, and the expected result of applying this correction factor, are shown in the graph on the right.

#### Looping modes

Progression from one iteration to the next happens in one of two ways, depending of the value of **Auto-loop** in the configuration file (see Section 1.1.3). If this is set to False, the user will be asked after each iteration to choose the gain for the next iteration, and continue (or otherwise) when ready. This is done via a dialog box. **Apply** will launch the next iteration. **Replot** recalculates the correction and the next AWG trace without applying, and shows the updated plots.

The **Pause** will close the dialog box without applying any correction, and provides the opportunity to inspect the graphs more closely using the graph toolbar to zoom or pan across any the graphs. The window can also be saved as an image. To reset the views use the home button. Once finished with inspecting the graph, the loop may be relaunched with the **Continue** button, or terminated by simply closing the loop window.

When **Auto-loop** is set to True, the user does not have the opportunity to change the gain between iterations, or inspect the graphs while looping. Instead, the dialog box above is replaced with a countdown timer until the next iteration begins, and the **Continue** button in the loop window is replaced with a **Stop** button. Clicking either the **Cancel** button in the countdown timer or the **Stop** button will abort the loop. If data is being written to the AWG, the loop will not abort until this process is finished.

In both cases, the loop will automatically terminate when any of the following conditions are met:

* RMS error < Tolerance
* Peak power > Safe value
* Iteration number = Max. Iterations

At the end of a loop, close the loop window to return to the setup window, from where a new loop can be configured and launched. The text window at the bottom of the loop window provides some basic information about the status of the loop after each iteration.

### Configuring the software

A text file file, **config.ini,** provides for configuration of the behaviour of the software. The options are described in Table 1:

|  |  |  |  |
| --- | --- | --- | --- |
| Section | Name | Description | Allowed values |
| file\_locations | diag | Location to save the diagnostic files | Valid file path |
| file\_locations | curve | Location of files in the curve library | Valid file path |
| safety | pulse\_peak\_power | Safe level of peak power | Float |
| safety | auto\_loop | Flag to switch auto-looping on/off | True/False |
| timing | auto\_loop\_wait | How to long between each iteration in auto-looping mode in seconds | Float |
| timing | scope\_wait | Time (in seconds) to allow for each acquisition from the oscilloscope. Recommended to leave at the default (2.1) | Float |
| timing | awg\_wait | Time (in seconds) to wait after writing each value to the AWG in point-by-point mode. Recommended to leave at the default (0.5) | Float |
| pvs | scope | Default value for the feedback oscilloscope PV | String |
| pvs | awg | The root of the AWG PVs | String |
| sim | sim | Flag to turn on/off simulation mode | True/False |
| awg | awg\_zero\_shift | When the the current output is zero, but the targets is not, the loop can’t converge as zero multiplied by any number is 0. If this is detected, this value will be written to the AWG on the first loop to allow convergence. | Float between 0 and 1 |
| awg | awg\_ns\_per\_point | Time calibration of each sample of the AWG. Should not be changed. | Float |
| awg | awg\_write\_method | Either write to the AWG is point-by-point mode or waveform mode | pts/wfm |
| epics | epics\_ca\_addr\_list | The address list searched for EPICS communication | Valid list of IP separated by commas |
| epics | epics\_ca\_auto\_addr\_list | Whether to use the EPICS auto address list | YES/NO |

Table 1: Contents of config.ini configuration file

After editing this file, the software must be restarted for changes to take effect.

### Writing to the AWG

The software can write to the AWG in one of two ways: point-by-point, or waveform. In the former case, the calculated values are applied to the AWG one point at a time. On the first iteration every point of the AWG will be checked to see if a write is necessary to ensure that values outside of the specified pulse width are set to zero. In subsequent iteration, only number of AWG samples for the requested pulse length are affected. This makes this method slow for long pulse duration, but fast for shorter values.

In waveform mode, the entire waveform is sent in one go. Every point of the AWG is written at each iteration, and therefore the time taken for this operation does not increase with increasing pulse length. This may also have the advantage of avoiding transitory spikes in the AWG trace that can sometimes occur when writing a pulse in point-by-point mode. Note: no safety checking takes place while writing to the AWG, only beforehand to check that the final state will be safe.

As a rough guide, using default values, the point-by-point method will be faster for pulses shorter than around 5 ns, above this the waveform method is quicker.

### Simulation mode

It is possible to test much of the functionality of the software, and gain familiarity with the interface, without connecting to any hardware by setting the **sim** flag in the configuration. This software will then show simulation in the title bar of the setup and loop window. In this case, you can configure and run a loop to see how the interface works. It is still necessary to specify a background file and a target curve but it not necessary for the **Scope PV** to be connected.

### Format for files

All the files should be in text format, with one value per line.

The background file must have the same number of data points as will be present in the acquisition from the feedback oscilloscope. This should be the case automatically if the background is taken with a **Grab**, but if for some reason a synthetic background is used this has to be accounted for. The default number of points is 1000.

Target files are text file containing values between 0 and 1 that represent the normalised temporal shape of the desired pulse. The number of values is not critical – it will be resampled over the correct of AWG samples for the desired pulse duration – but should contain enough points to correctly describe the pulse. By saving these curves with a .curve extension into the location of the curve library folder, these files will appear in the predefined drop-list at start up. The name which appears in the list is the filename without the .curve suffix. For example my\_new\_curve.curve would appear as my\_new\_curve in the list.

### Setting the Start Point

TODO