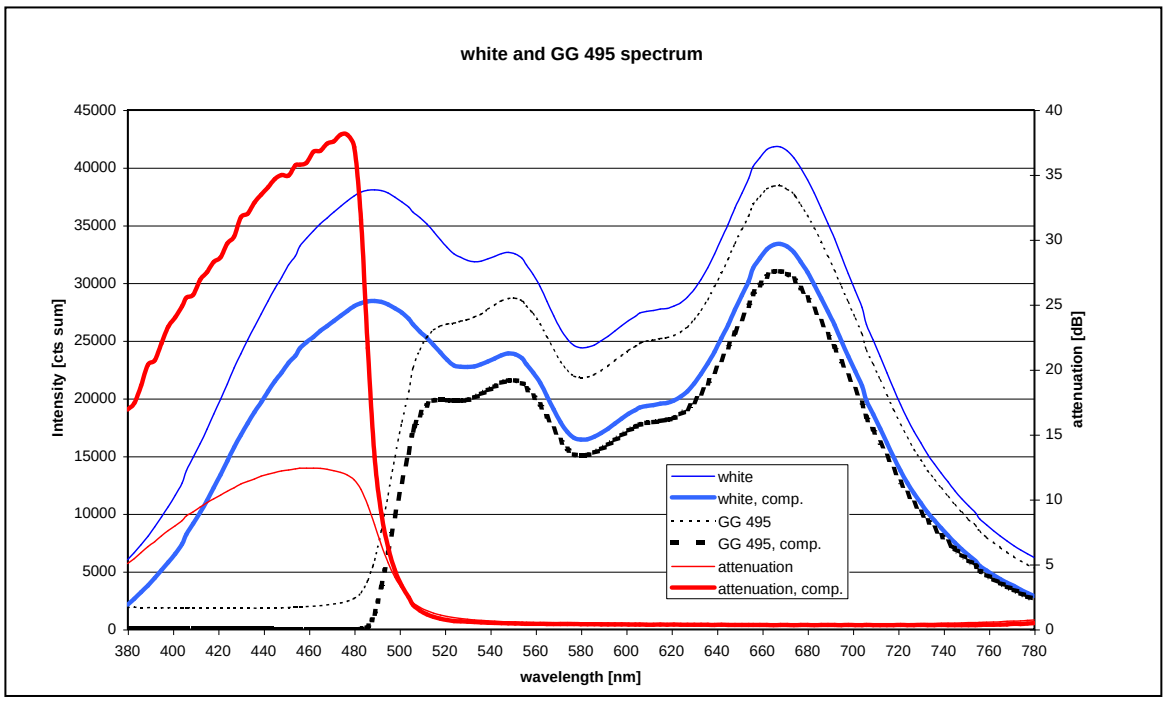
1. Introduction

Stray light is defined as an electromagnetic radiation of a given frequency which is finally not imaged onto its respective wavelength position on the detector array. Reason for stray light effects in the Microspectrometers are a minor surface roughness caused by the molding process to produce gratings. Stray light limits the photometric accuracy in certain applications. As the reason for the stray light is inherent to the device specific surface finishing of the grating – which remains constant over the lifetime of the device – SC30 uses a mathematical model for the compensation of the stray light.

The SC30 based on the conversion of the measured spectrum through the transfer-function of the spectrometer-system.
The system-transfer-function is a square matrix that has been calculated by monochromatic stimulation for each mould insert. Each spectrometer is then supplied with this matrix which is called data set.
During the measurement the data set transforms the measured spectrum into a stray light compensated spectrum.
The diagram below illustrates the effect of SC30 on a cut off filter GG495. Without using the SC30 a maximum absorbance of approx. 14dB can be achieved. With SC30 a maximum absorbance of approx. 35dB can be achieved with the same filter.
An improvement of the spectral resolution during measurements of narrow-banded samples is achieved through the compensation of the close field stray light. This compensation also takes into account the higher order effects.



1.1. Description of the SC30 parameters

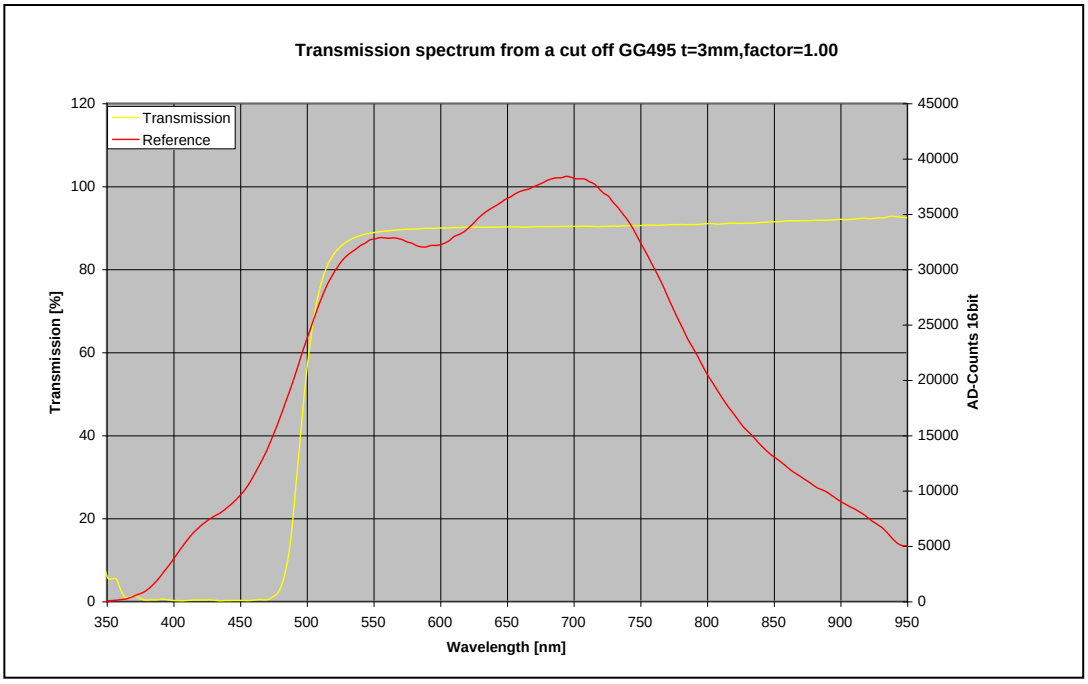
Correction factor:

This factor defines how many stray light will be add out from the measured spectrum. The factor depends on the use light source and has to be found over an experimental way. In the list below you find some factors, as reference points, for different light sources:

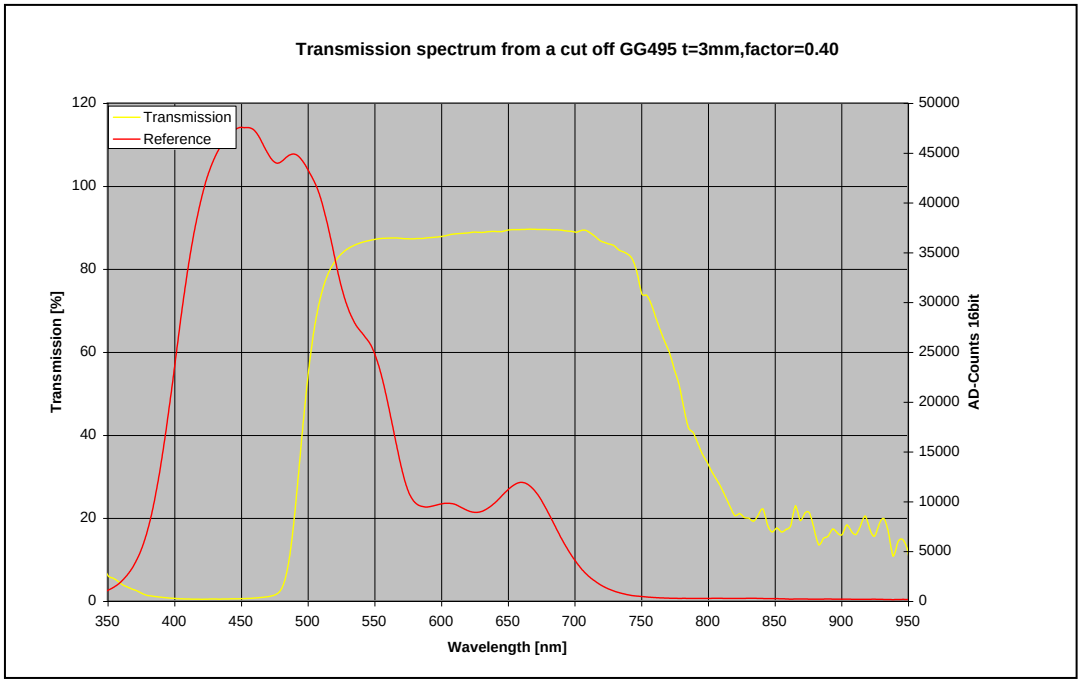
Light source:	**Factor:**
Tungsten-halogen 2800K	1.00
Tungsten-halogen 3300K	0.50
Tungsten-deuterium	1.30
Deuterium	0.80
Xenon flash lamp	0.60

These factors were determined with a cut off filter type GG495 thickness 3mm. In the following diagrams the transmission spectrum of the filter and the intensity spectrum of the corresponding light source (Reference) are shown.

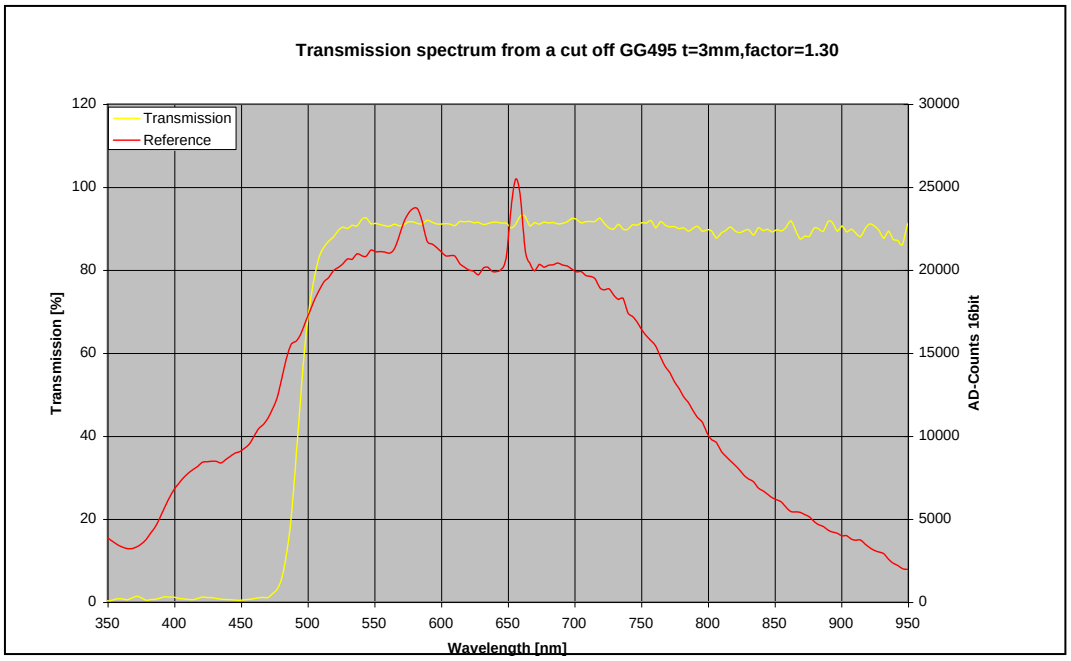
Tungsten-halogen 2800 K, factor 1.00:



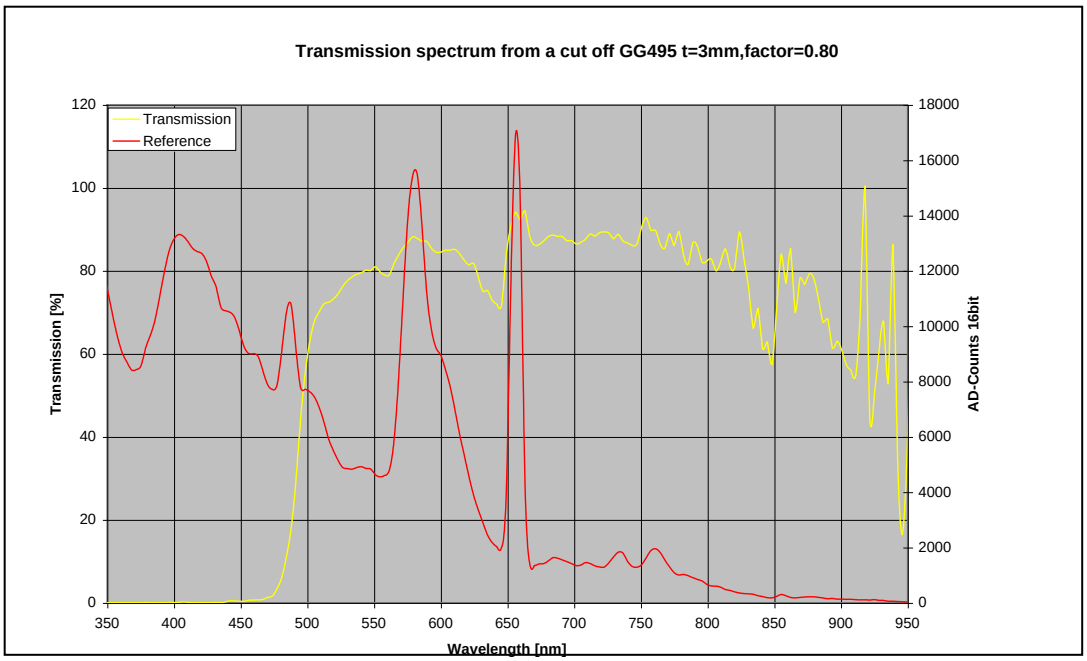
Tungsten-halogen 3300 K, factor 0.40:



Tungsten-deuterium, factor 1.30



Deuterium, factor 0.80



Dark spectrum:
A value array which has been readout from the diode array without any light influence the diode array.

Reference spectrum:
A value array which has been readout from the diode array, which included the spectral information's.

Spectrum:
A value array which has been calculated out of the difference from reference and dark spectrum.

Number of values:
The numbers of stray light reduced values that has been returned by the SC30.

2. Using the SC30 in your applications

For the simple binding of our SC30 into your applications, two possibilities are available:
If you possess an OEM system, then you execute the SC30 with the commands they are implemented in the firmware of the electronic. This firmware commands are described in the following **chapter 2.1**.
For all other applications our *SC30_11.dll* is available, which you find on the provided CD-ROM in the directory SC30. You find a detailed description of the *SC30_11.dll* in **chapter 2.2**.

2.1. Using the SC30 in OEM Systems

In the OEM Spectrometer Systems you get an easy access to the SC30 over firmware commands:

Get correction factor
Returns the correction factor.
Syntax: *para:sc30corr?<CR>
Returns: String “SC30 correction factor: 1.50<CR>” (In this example the correction factor is 1.50).

Set correction factor
Sets the correction factor (Min = 0.1 and Max = 5.0).
Syntax: *para:sc30corr factor<CR>
Returns: ACK when the command was accepted and NACK when the command was not accepted.

Get number of values
Returns the number of values which will be sent from the SC30-Algorithm.
Syntax: *para:sc30pix?<CR>
Returns: String “SC30 pixel count: 208<CR>” (In this example 208 values will be sent from the SC30-Algorithm).

Start SC30 calculation
Starts the calculation of the SC30-Algorithm with the last measured reference spectrum.
Syntax: *calc:sc30<CR>
Returns: NACK when the calculation could not started. Other wise x number of values will be returned in the format <wavelength><TAB><CorrValue><CR>.

NOTE: The *x* number of values are the numbers that will be returned by the command **para:sc30pix?*.
Before a SC30 calculation could be successful started, a dark and a reference spectrum has to be measured with the commands:

T *tint av* <CR> or *meas:dark *tint av format* <CR> for a dark measurement.
R *tint av* <CR> or *meas:light *tint av format* <CR> for a reference measurement.

2.2. Using the SC30 in the SC30_11.dll

Calling convention is **__stdcall**

Remark:
The "calling convention" for the dll is defined by the compiler which created the dll. The most languages, for example Visual Basic, Delphi and so on, prevent only the __stdcall calling convention.
C++ programmers must place the **__stdcall** convention in front of their function declaration.
For example:

```
extern "C" __declspec(dllimport) int __stdcall SC30calculation (float
*Spectrum,int LengthSpectrum,char* Filename,float CorrFactor,float
*CorrSpectrum);
```

Call up function:
The function is a C++ function where

float is defined as 4 byte with the range $3.4E^{+-38}$ (7 digits)

int is defined as 4 bytes with the range –2.147.483.648 to 2.147.483.647

*char** is defined as a pointer to a *string*.

The syntax of the function is:

```
int SC30calculation (float *Spectrum,int LengthSpectrum,char*
Filename,float CorrFactor,float *CorrSpectrum)
```

The alias name of the function is:

```
_SC30calculation@20
```

Use the „alias" for programming languages they don't supported a direct access over the function name.

Example for a call up in Visual Basic over the alias:

```
Private Declare Function Sc30Calc Lib "C:\Temp\Sc30_11.dll"
Alias "_SC30calculation@20" (ByRef Spectrum As Single, ByVal
length As Integer, ByVal Filename As String, ByVal Factor As
Single, ByRef CorrSpectrum As Single) As Integer
```

Description of the parameters:

Filename:
Include the file path and file name of the data set.
For example: *"C:\temp\sc30.dat"*
You will find the data set on the provided CD-ROM in the directory "SC30". The data set has the file extension *".dat"*.

LengthSpectrum:
Included the length of the spectrum array.

Spectrum:
Adapted array which corresponding with the wavelength range of the data set.

Because the data set is limited to a wavelength range, the measured spectrum has to be adapted to this range.
For this, you have to cut out your spectrum between "1st SC30 pixel" and "lst SC30 pixel", when your spectrum is longer than the wavelength range of the data set.
Respectively, when your spectrum is smaller than the wavelength rang of the data set, you have added dummy values into your spectrum till "1st SC30 pixel" and till "lst SC30 pixel".
The "1st SC30 pixel" and "lst SC30 pixel" can be finding on the provided data sheet of the spectrometer. The counting of "1st SC30 pixel" and "lst SC30 pixel" begins at position ZERO!!

In the table below you see a extract data set (red and grey region in the table), which included the stray light information of the spectrometer. Also you see a spectrum (yellow region in the table). This spectrum starts at wavelength 318.44nm and ends at 1072.34nm. To adapt this spectrum to the wavelength range of the data set, you have to delete the first three and the last two values in your spectrum. Because the *SC30_11.dll* ignores values which over end goes out and to prevent the *SC30_11.dll* error *-3*, adds three values more at the end to your spectrum than absolutely necessary.
After the *SC30_11.dll* has been run successful, you get back a value array of stray light reduced values called "CorrSpectrum". The length of this "CorrSpectrum" is equal to the length of the parameter "Spectrum". For the values in the spectrum which over end goes out, the *SC30_11.dll* insert zero values into the "CorrSpectrum". This zero values can be ignored.

Index	0	1	2	**3**	…	**211**	212	213	
				1st SC30 pixel	…	**last SC30 pixel**			
Wavelength	318.44	322.01	325.59	329.16	…	1065.53	1068.94	1072.34	1
Spectrum	851	884	930	972	…	61	63	71	
			Wavelength	330.00	…	1064.00			
			330,00	9635	…	29			
			336.00	2119	…	35			
			343.00	194	…	21			
			350.00	109	…	25			
			357.00	75	…	11			
			364.00	58	…	12			
			371.00	53	…	26			

Please always look for an example on the provided CD-ROM in the folder "\SC30\SC30dll_Example\". This example is an illustration in an Excel worksheet with a Visual Basic Macro. In this example you can see how you adapted the spectrum and how you can call the *SC30_11.dll*.
Please read first the "ReadmeFirst.txt" file in the same folder!!

CorrSpectrum:
Array that included the stray light corrected values.

Return Value:

Type	Description
Integer	0 SC30 calculation succeed -1 CorrFactor outside of range -2 No data set found -3 Spectrum too small for the selected data set

System requirements:
PC with operating system Windows 2000/Windows XP/Windows Vista and Windows 7.

Installation:
No further installation is required.