Calculating an unbiased fluorescence decay curve recorded by a TCSPC system with detector and electronics dead-times

The function **`TCSPC_DTcorrected`** calculates decay curves measured with a single-photon counting system with ps/ns temporal resolution under pulsed excitation. The function returns both an unbiased, dead-time corrected TCSPC curve as well as the uncorrected "standard" TCSPC curve. The function call is:

```
[tcspc_corrected,tcspc_standard,epsilonP] =
TCSPC_DTcorrected(tPhoton,syncPeriod,DT_electronics,DT_detector,flag)
```

All input parameters except for **`flag`** are in time-units of TCSPC bins.

`tPhoton`	A vector containing the absolute photon arrival times.
`syncPeriod`	A scalar containing the time between two consecutive sync pulses (P in the paper).
`DT_electronics`	A scalar containing the dead-time value of the electronics (E in the paper).
`DT_detector`	A scalar containing the dead-time of the detector (D in the paper).
`flag`	If set to 1, the folder "subroutines" has to be added to the MATLAB path manually (see below).

The outputs **`tcspc_corrected`** and **`tcspc_standard`** of the function are the unbiased and biased decay curves, respectively. The former is normalized such that **`sum(tcspc_corrected)`** equals the average number of photons per excitation period (εP in the paper), the latter is not normalized, therefore **`sum(tcspc_corrected)`** equals **`numel(tPhoton)`**. The output **`epsilonP`** is the average number of photons per excitation period εP. It can be used to obtain a corrected intensity value for the pixel by multiplying with the number of excitation cycles for this pixel via:

```
intensity=ceil( (max(tPhoton)-min(tPhoton)) / syncPeriod) * epsilonP
```

Note for improving performance:

In order not to clutter your file system, we put all subroutines into a folder called "subroutines", which is added to the MATLAB path in the first few lines of **`TCSPC_DTcorrected`**. If you call this function several times, for example in a script where you evaluate many pixels, there is no need to add the subfolder to the path every time. In this case, we recommend that you set **`flag`** to 1 and add the folder manually / in the script that calls **`TCSPC_DTcorrected`** using the following code:

```
fullPathToThisFile = mfilename('fullpath');
[path,~,~] = fileparts(fullPathToThisFile);
addpath(genpath([path,filesep,'subroutines']));
```

Example:

A typical example is shown below. The photon arrival times were simulated for a TCSPC bin width of 64ps, εP = 1.5, an excited state lifetime of 3ns, an excitation period of 50ns, an electronics deadtime of 80ns, a detector deadtime of 30ns and a total number of 5000 photons. The photon arrival times are given as the vector **`tPhoton`** in the file **`tcspc_example.mat`**.

```
syncPeriod = 782;
DT_electronics = 1250;
DT_detector = 516;
load('tcspc_example.mat');
[tcspc_corrected,tcspc_standard] = TCSPC_DTcorrected(tPhoton,…
      syncPeriod,DT_electronics,DT_detector);
t=0.064*(1:782);
plot(t,tcspc_standard/sum(tcspc_standard),'-b',…
      t,tcspc_corrected/sum(tcspc_corrected),'-r');
xlabel('time [ns]')
ylabel('normalized intensity [-]')
legend('standard TCSPC curve','unbiased TCSPC curve')
```

A simple exponential fit yields decay times of 1.64ns and 2.93ns for the biased and unbiased TCSPC-curves, respectively. As expected, the unbiased curve gives a much better result.