PicoHarp Questions & Answers

Principle of Operation and Common Setup Questions

> How does the PicoHarp work? Can it replace a correlator card ?

The PicoHarp basically does this:
Repeatedly measure the arrival times of events on both input channels with picosecond precision. In Histogramming Mode it then calculates the time difference between subsequent start and stop events and puts the results in a histogram.
It the Time Tagging Modes it continuously delivers individual photon records to the host PC. In case of histogramming, the result is related to, but not identical to a correlation in its mathematical sense.
In a true correlation one would use ALL time differences between ALL possible combinations of start/stop pairs. In Histogramming Mode one only obtains a subset of that. The result still reflects the correlation function, only that it is limited to the zero order time differences and we used only a small fraction of all emitted photons. Nevertheless this method delivers very useful results and is commonly used in photon coincidence correlation experiments.
By means of the Time Tagging Mode T2 it is possible to even obtain the full correlation function with picosecond resolution and virtually unlimited lag time range. Both is impossible with classical correlator cards (e.g. from MALVERN or ALV) that can do a true correlation but only with some nanoseconds resolution and limited lag time range. The difference is that with the Picoharp the correlation must be calculated in software. With modern computers and efficient algorithms this is not a serious limitation. Under certain conditions it can even be done in real-time.

In practice the PicoHarp can be used in many different ways. The mainstream use is in Histogramming Mode with a laser sync signal connected to the start input (channel 0) and a photon detector (SPAD, PMT) at the stop input (channel 1). This allows e.g. time-resolved fluorescence measurements upon pulsed laser excitation. Another application is coincidence correlation as mentioned above (e.g. for antibunching measurements). In this case both input channels are connected to photon detectors. With the Time-Tagged Modes one can not only investigate correlations but virtually all other aspects of the photon dynamics. This allows extremely sophisticated methods of data analysis e.g. in single molecule spectroscopy or quantum communications.

> What is the difference between the TimeHarp 200 and the PicoHarp 300?

The TimeHarp 200 is a PCI card and has a time resoluion of about 35 ps. It is the proven workhorse used in hundreds of installations world wide. It was particularly successful in single molecule spectroscopy. For this area of application and with detectors of limited time resolution it is still a very good choice.
The PicoHarp 300 is the more recent product and provides more functionality and higher time resolution (4 ps). As opposed to the TimeHarp it is not a PCI card but a

stand-alone box that gets connected to the PC via USB. The PicoHarp is recommended today if it is desired to make best use of high resolution detectors such as MCP-PMT or the new MPD SPAD detectors. Similarly, it should be chosen if the advanced Time-Tagging modes of the PicoHarp are required for the application.

> I am trying to understand the specifications of deadtime, time resolution and
> maximum count rates. I would like to compare them with other instruments but
> the specifications are partially using different terms. Specifically I am looking at
> the counting board (PCI card) Model P7887 by the company Fast Comtec.
> This board is capable of accepting one event (stop pulse) in every time bin.
> The time resolution is 250 ps. They say burst/peak count rates of up to 4 GHz
> can be handled with no dead time between time bins.
> The PicoHarp 300 has a much better time resolution of 4 ps but the maximum
> measurement rate of PicoHarp300 is up to 10 million cps. Is this coming
> from the 90ns dead time of the PicoHarp 300 ?.
> What is the different between P7887's 4 GHz and PicoHarp's 10 Mcps?

You need to distinguish between the "burst" count rate and the "continuous" or "sustained" rate. All such instruments have a certain deadtime which results at least from the "pulse pair resolution" because they need minimum pulse width plus some time for a gap between the pulses. In the case of the P7887 this happens to be the same as their time bin width but that's just the specific design and need not be so for all such instuments. Picosecond timing as performed by the PicoHarp requires more complicated processing so that the deadtime is longer. The 90 ns for the PicoHarp are actually the shortest you can get today for such a resolution. Now, if the instrument can process a few consecutive counts in the tightest possible sequence then the burst count rate is simply 1/deadtime. These are the 4 GHz specified for the P7887. On the other hand it is impossible to constantly process and store 4 Gig events per second. This is why the continuous count rate must also be looked at. In their case it is specified as 12.5 MHz (in some document it says 11 MHz). The capability of handling the high burst rate is typically implemented by fast FiFo buffers. Due to their limited size, the high rate can only be handled fo a few tens or hundreds of counts in a row. Dependent on the application this can make sense if the avarage rate is much lower than that of some occasional bursts. In case of the PicoHarp the burst rate is given by 1/90 ns which results in 11.1 MHz. For simplicity we specify it as 10 MHz, which is the SAME as our continuous rate in on-board histogramming. In TTTR mode our continuous rate is 5 MHz due to the USB throughput limitations. It is worth noting that a pulse pair resolution or deadtime of 250 ps is rarely useful with real-world detectors. They usually have deadtimes on the order of tens of nanoseconds. Similarly, this limits the usable maximum count rate so that the Picoharp is perfectly capable of handling them.

> What is the significance of „multi-stop“ as opposed to one start one stop?
> If the PicoHarp 300 has multi-stop capability, does that mean it can support
> burst/peak count rates of up to 250 GHz with no dead time between time bins?

No. The multi-stop capability is still subject to deadtime. It therefore is really only of interest at very low start (sync) rates as is the case in classical fluorescence or luminescence lifetime measurements when performed e.g. with UV flash lamps (some KHz). Then the Picoharp is of huge advantage over a TAC based system because it can measure hunreds of times faster. A TAC can only process one stop per start. In classical fluorescence/luninescence lifetime measurements with TCSPC this leads to capturing always the early photons and dropping the late ones. This results in a skew of the resulting histogram. The only solution to keep this distortion low is to work with slow count rates. The rule is to use ~1% of the sync rate. This is why in the case of flashlamps above the usable count rate gets painfully slow. Btw this is why we offer lasers with rates in the MHz range. If someone cannot use those, a TDC based system with multi-stop capability (such as the PicoHarp or NanoHarp) can go much faster because it does not prefer the early photons. On this matter, please read also our Tech note on TCSPC. It is on your software CD under "techdocs".

> Can the PicoHarp be used together with cards of manufacturer XY or do they
> possibly cause conflicts when they are installed to the same PC?

The PicoHarp 300 is a USB device, which is handled by the Plug&Play operating system of the PC. This means that resources are automatically allocated and shared between such devices. Conflicts wih other devices that cause complete failure are not to be expected. However, the USB has a limited bandwidth that must be shared between the devices connected to it. If the PicoHarp is used in Time-Tagging Mode it must have sufficient bus bandwidth available in order to transfer the incoming data in real-time. Other bandwidth demanding USB devices should therefore not be operated together with the PicoHarp.

> what does the "calibration button" in the PicoHarp software do?
> Do I need to disconnect any cables from the board before I turn on the software?
> In other words, is the "calibration" the software executes when I turn on the
> software affected by the existence of external cables connected?

The calibration is done internally. When calibrating, all cables can remain connected. Of course, when you calibrate you cannot measure but it only takes a few seconds to calibrate. You only need to calibrate one time before starting your measurements, after a warming up period of ~30 minutes.

> How can I adjust the delay time between the detector signal and the sync?
> Except for changing the cable lengths, can the software do this adjustment?

You need to change the cable lengths. Currently there is no other way to do this. There are also switchable delay boxes e.g. from Ortec if you need to change your setup frequently. Proper delay settings are important.
The ‘Offset’ parameter in the software only shifts the measured range of start-stop time differences so that they come to lie in the acquired histogram. It should usually be zero. Only when operating at high resolution but with very long time differences this offset can be useful.

> Using on-board histogramming, what is the number of events that can be
> stored before emptying the PC memory? Is this automated?

Each histogram cell is 16 bits, so 65535 counts per channel. It is automated in so far that you can configure it to stop when one channel is filled to a given level (StopAt). In practice this does not happen so quickly due to limited count rates.

> How to set the CFD levels (zero cross and discr.) correctly? I found
> that at different set values, you can get some different fluorescence decay
> curves with a bit different FWHM. Question is which one is correct and how
> to judge?

There is no general rule to this. The goal when adjusting this is that usually one wants a narrow IRF but also preserve as many counts as possible.
A good way to start is to look at the count rate you get at CFD discrimination level = 0. If you see huge count rates, go a little higher. The Zero cross is not critical at this point. Set it to 10 mV.
Then you increase the CFD discrimination level until you see an improvement (reduction) in the IRF width, probably also some noise suppression. At the same time you check that the count rate does not drop too far.
Typically one works at about 70% of the undiscriminated rate. 50% should be the limit. Then you can try to modify the zero cross level for further IRF improvement.

> We use the Hamamatsu H5783 as the detector, so far I measured the IRF
> being about 0.66 ns wide (FWHM). Is this value reasonable, or it is too big? It
> also depends the CFD level set values.

Assuming your laser pulses are ok, this is pretty much. One can get down to 180 ps with these detectors, provided the laser pulses are narrow and the setup is correct. Make sure you run the PMT at the right voltage. The highest usually works best, although it stresses the detector more. The IRF also depends on the area you illuminate. A small focus often improves the IRF. It surely depends on the CFD level. That should be used to optimize the IRF width while still not discriminating out too many counts (max. 50%). Your sync detector also influences the IRF. Make sure it is illuminated properly (lens?) and try to vary the sync level and watch IRF changes. Generally make sure you have no stray (scatter) light in your optical setup. If you use a scattering solution (e.g. Ludox) in a cuvette, make sure it is not too highly concentrated. If it looks like milk it is too much. The turbidity should be just about visible. If your laser is not very stable it can also improve your IRF if you make sure to trigger on the same laser pulse that your start input is looking at. This requires to find the right amount of sync delay (usually a few meters of cable).

> Why do I see a difference between the input rate, the histogrammer count rate
> and the "Total count rate"? In all my measurements the histogrammer count rate
> is about twice as high as the Total count rate I do not understand this difference.

Here is probably a misunderstanding about "Total count rate". There is only a "Total Count" which is just the sum of all counts in the histogram. So it is not a rate! You probably see the factor of two because you measured for 0.5 sec. The input rate is what the discriminated input by itself sees. The histogrammer count rate is the number of usable start-stop-pairs per second that were actually processed in the histogram. Indeed the histogrammer rate should be approximately the same as the input 0 rate, unless you have very low sync rates. Then the probability of getting usable start-stop pairs is smaller and the histogrammer rate drops accordingly.

> I tried to monitor the signals at the PicoHarp inputs with an oscilloscope. I used a
> T-connector to attach the scope. What I see is double pulses or pulses with
> strange humps and the PicoHarp shows different meter readings and histograms
> than I see without scope connected or sometimes the PicoHarp shows no signals
> at all.

You definitely cannot use a simple T-Connector. All signals must be 50 Ohms terminated and free of reflections. When you attach two 50 Ohms loads to one source you have improper impedance matching. The only way to achieve this is to use a "power splitter" (aka reflection free T-pad). These items are available from PicoQuant. Note that a power splitter will give you only 50% of the original pulse height at each output branch.

> With uncorrelated light we always get some almost constant amount of photons
> detected along the time axis and it suddenly goes to zero at around the end of
> time axis. Is this supposed to be like this? Or is there something wrong?

The time axis limits can be set to arbitrary values. This may not correspond with the actual measurement range available at the current resolution or the time span between two sync pulses. Assuming a fixed sync period, the histogram can never be wider than the time span between two sync pulses. Therefore it is quite normal that your histogram ends abruptly at this time span. If it ends earlier, it may be the limit of the histogram range given by your chosen resolution and the maximum number of histogram bins.

> I am using SPAD detectors SPCM-AQR. I get about 500 ps FWHM which I would
> like to improve. When I try to optimize my IRF it seems that the CFD level and
> Zero Cross are not having any influence.

The CFD is primarily designed for PMT detectors, where different pulse heights may occur. The SPCM-AQR have pulse shaping electronics built in and it is rather pointless to try and optimize the CFD for these signals. In this case the CFD becomes a simple level trigger and the CFD level should be set somewhere at half the pulse height. The SPCM-AQR have a specified time response of 350 ps but in practice it is rarely better than 400 ps which is near to what you observe and what we have measured ourselves. The contribution of the PicoHarp to the total IRF is very small since the individual contributions add up geometrically. The critical point is usually the optical focus. You can try to illuminate a smaller area in the center of the detector, but do not expect anything better than 400 ps. Another issue is count rate. These detectors get worse with increasing count rate, so you can try to go slower. They also show a significant shift in pulse position with changes in the count rate. So if your count rate fluctuates strongly, there will be some IRF broadening.

> We are setting up our time-resolved system with a PicoHarp 300.
> When looking at the laser light (5ps mode-locked Ti:Sapphire at 80 Mhz

> rep. rate), we see a strong peak which is asymmetric with an exponential
> tail on the right and with a FWHM of about 600 ps. However, the resolution
> should be limited by the APD (350 ps) and we expect a symmetric peak.

Indeed you should get a narrower peak but it is not necessarily symmetric. Care must be taken to illuminate the APD only in its center (possibly through a pinhole).
If you illuminate the whole active area your IRF can get even broader than 600 ps. It is hard to achieve the specified 350 ps.

>We use an APD with the PicoHarp and tested the IRF with a scatter solution.
>The scatter traces look quite asymmetric, and the width is a bit concerning me,
>it is 750 ps, which is quite a lot above the specs of my APD (300 ps).

Indeed 750 ps is very broad. It may be that you are actually seeing a fluorescence decay. You should check what materials there are for scattering. May be they fluoresce? Are you measuring through colour filters? That might favour fluorescence light over scatter light. The SPCM-AQR detectors also get worse with increasing count rate. They also show a significant shift in pulse position with changes in the count rate. So if your count rate fluctuates strongly, there will be significant IRF broadening beyond the specs. The more recent models have slightly improved in this matter but it is still showing.

> You mention in your documentation that the dead time is <95 ns, what
> happens IF/WHEN an event arrives during that dead period, would this impair the
> measurement?

Those signals would simply be ignored. This is no problem. If you imagine photons originating e.g. from a single molecule, they would radiate out in all directions and your detector only "sees" a small fraction of them. In this sense it doesn't matter if in the analysis drops another few. It simply appears as if they had never reached the detector. In fact, the detector's own deadtime has the same effect. The only important condition related to dead time is that your photon flux is suitably attenuated, so that statistically only approximately one in 100 laser pulses generates a photon. If you disobey this, your histogram gets distorted by so called pile-up but possibly also due to deadtime effects.

> Also the specified max count rate in histogramming mode is 10 MHz. This, I
> guess, depends on the processsor or the PC.

No, it is actually just the approximate inverse of the dead time. In histogramming mode it does not depend on the PC because the timing is done by dedicated hardware in the device.

> Is there a min/max pulse signal duration for the detector
> channel? For instance we would be using the PicoHarp 300 with an
> avalanche photodiode (EG&G) giving signals of 25 ns, with TTL signal that seem
> to be incompatible with your voltage specs, first, do you have adaptors for
> that? What consequences would there be if we'd use these detectors?

The PicoHarp is designed to work with these detectors. It triggers on the signal edges and pulse widths up to 50 ns are fine. All you need is a suitable pulse inverter because the PicoHarp needs negative input signals. This inverter/attenuator module (SIA 300) is available from PicoQuant.

> What is the differential non-linearity (DNL) of your ADC? Becker and Hickl talk
> about some error correction coding they do to get in the 1% range
> (dithering etc). I have heard that there may be artifactual ripple in the
> lifetime histogram for flash ADC implementations.

Indeed their error correction concept is needed due to the bad linearity of fast ADCs. It is ok to use, but it limits the usable time range on both sides. In real lab practice we have never seen anything better than 3% with their cards (assuming that you mean peak-peak). For the figures they specify, they use an artificial testbed with two function generators which is very different from what you have in reality. For their DNL they also seem to subtract out the theoretical random noise contributions, so that (theoreically) only the systematic part is specified. We only test in a real setup with picosecond diode laser and PMT and specify the data accordingly (PDL 800B, 635nm @20MHz and Hamamatsu PMT H5783 exposed o daylight). We also do not subtract out random noise components. The figures are therefore not directly comparable.
The technology we use for time digitization in the PicoHarp is very different, so the B&H correction scheme is not needed. In the real world testbed described above, our PicoHarp shows a DNL of typically 3%p.p. and <0.5% r.m.s. This is identical with what we found in our own measurements with B&H cards in the same testbed. With the typical DNL of either product you should get excellent results in the data analysis. If you do observe problems, inherent time digitizer DNL is rarely the root of the problem. Much more critical in real life are interferences between unshielded detectors/lasers or noise pickup through ground loops etc.

> We use a PicoHarp 300. Installation was OK and the sync is detected well
> (80 Mhz from Ti:Sapphire laser). We connect the APD (EG&G avalanche
> photodiode through a SIA 300 inverter to get the right pulse polarity. The APD is
> installed in a very light-shielded box. When we run in oscilloscope mode, we get a
> few hundred counts/s that increase when we increase the ambient light level.
> Changing the CFD level adjust and/or CFD zero cross adjust does not change
> anything to the look of the counts (at least so it seems). Does that indicate that
> something is wrong with our connections/setup of the system?

No. The EG&G or Perkin Elmer APD modules deliver TTL pulses which have always the same shape. The SIA 300 just inverts and attenuates them. The CFD therefore has "nothing to do" in terms of fluctuating pulse heights. It is operating as a simple level trigger in this case, which makes it very insensitive to level and zero cross changes. That's just fine. From your description it is not clear if the light on the detector was that from the laser. If it was not, you will not see much influence of the CFD settings anyway. In order to optimize your instrument response it is necessary to use the laser light (strongly attenuated). In this case you may try to find an optimum in the CFD settings where the IRF is narrowest.

> We are currently using your PicoHarp 300 for TCSPC applications.
> We use a confocal microscope with excitation from a Red-diode laser
> (Rep rate = 40 MHz; pulse width ~50-100 ps).
> The SYNC pulse is delivered from this laser. Detection is done with a single
> photon avalanche photo-diode (Perkin Elmer SPCM-AQR-14-FC9700). The
> detector is connected to the PicoHarp through a SIA 300 inverter/attenuator.
> With this setup, we achieve IRF's around 400 ps.

OK, this is quite a typical figure. Nominally these detectors are supposed to be better but they never really are. You also need to focus very tightly.

> Another question is this: If I change the relative position of the
> SYNC and signal pulses with cable delays, this window does not change, but
> the signal is scanned through this window. If large cable delays are
> used, I can push the signal to the very edge of the window, and then it
> starts to wrap around as I continue to add delays. Why is this?

This is perfectly normal. The timing is always with respect to the last sync pulse, whichever it is. Since the histogrammer only gets the time difference it won't know. So if you delay e.g. the detector signal, the peak will move to the right and at some point snap back to the left edge when it has passed the next sync pulse. You need to put it in a position that allows you to fit your entire decay in the window. If it is too large you need to reduce the laser rate so that you get a longer period.

> In our TCSPC curves we see the main decay peak centered around
> 10 ns, but there is also another peak, centered around 13 ns. No matter
> what we do to our system, this secondary peak is present. The seconday
> peak's shape also roughly correlates to the shape of the main peak. Do
> you have any insight into what this could be from?

This looks like electrical or optical reflections. Try moving optical parts and changing cable delays to narrow the possible source down. Check if it responds to discriminator settings. Make sure you have all electrical signals properly 50 Ohms terminated and no 'branches' in your cabling.

TTTR Mode

> What is TTTR mode? Can you give a description in a nutshell?

TTTR means Time-Tagged Time-Resolved. It is a recording scheme, where we do not form histograms on-board but record every photon individually, with its complete arrival timing information. This provides ultimate flexibility in the data analysis, e.g. for single molecule burst detection, burst size analysis, or correlation (like FCS) to identify diffusion constants etc.

> The TTTR (Time-Tagged, Time-Resolved) recording of the events is listed as an
> option for the PicoHarp 300. Does this option affect the performance of the

> device in any other way than adding this feature?

The TTTR option consists of a firmware and software add-on. It includes two tifferent time tagging modes (T2 and T3). There is no change of performance other than the added capability of recording individiual photons as opposed to the standard on-board histogramming. The add-on can be purchased as an upgrade any time later.

> Someone recently mentioned something like time-stamping. I assume this meant
> TTTR mode, right? I understand this mode would correspond to noting the arrival
> time of each photon. What is in that mode the minimum resolution?
> Are there any deadtimes in this mode?
> If there were deadtimes, would this cause trouble/problems/bugs if a photon
> arrived over that deadtime?

Yes, TTTR is essentially time stamping (or tagging) of each photon.
Regarding resolution we need to distinguish the two time tagging modes (T2 and T3). In both cases, the highest resolution is 4 ps. The difference is that in T2 mode both input channels are processed equally and each event is recorded with a single time tag of 4 ps. This is designed for connecting a photon detector to each input. In T3 mode one connects a laser sync to input channel 0 and a photon detector to input channel 1. In this case the PicoHarp processes event pairs consisting of a photon and a corresponding sync. The time tag for this T3 event pair consists of two parts. One is the number of the sync pulse, the other is the time delay between sync and photon. The latter can again be recorded at 4 ps resolution.
Dead time is the same as in histogramming mode (~95 ns). A photon that arrives during the deadtime it is just not recorded.

> What is the maximum time between 2 photons in TTTR mode?

It can be virtually infinite. The bit formats of the TTTR records have a fixed width but if there are overflows, a special marker will appear in the data stream, which will allow the software to detect any overflow and mainatin a correct time axis, virtually forever.

> Do you have a conversion program for TTTR mode files to ASCII?

TTTR mode files store one 32-bit record per photon. There is demo code provided on your distribution disk showing how such files can be accessed. This demo code can also be used to implement a conversion to ASCII files. However, TTTR files can get very big and it does not make much sense to convert them to even bigger ASCII files. ASCII files would be some ten times bigger and they would still be of little direct use. In order to interpret and analyze the TTTR data you probably need to write some dedicated analysis software anyway. Custom analysis software should therefore be designed to read the binary format directly. In our demos we can only show the smallest common denominator, because the data analysis needs are very different for different users. Nevertheless, we can of course offer software development services for your specific needs. We also provide a powerful TTTR analysis package derived from our Single Molecule Microscopy software.

> I have been unable to open files recorded under TTTR mode with the FluoFit
> software. One of the demo executables, the Pt3demo.exe did not
> open anything. Apparently a DOS windows opens itself but nothing happens.
> I thought that the PicoHarp software could convert the files into a readable format.
> Can you tell me how to read *.PT3 files ?

TTTR data is not primarily designed to be used for direct decay analysis with FluoFit. It carries a lot more information than simple histogramm data and reducing it to that would mean loss of information. Because there is a lot of information in TTTR data and users can have vastly different ideas of how to use it, TTTR data is meant to be analysed by custom programs that users develop themselves. For this purpose we provide demo programs that show how to read these files but they do not perform a specific kind of data analysis. Because these programs are just starting points for custom programming, we expect users to take a look at the source code (which is provided) rather than writing secondary documentation for them. The demos are desiged to be run in a command line box, not just by clicking on them. The demos will print a short description of their usage when used with wrong parameters. In the case of T3demo.exe this is as follows:

```
Pt3demo infile outfile
infile is a binary PicoHarp 300 T3 mode data file (*.pt3)
outfile will be ASCII
```

This will generate a list of the recorded photon events but that is not designed for any other specific purpose, it is just a demo for reading the files.
In case you really only want to transform TTTR data into histograms, this can be done by the PicoHarp software. This is explained in the manual section about TTTR mode. It is done by selecting *.pt3 for the file type in the File Open menu. The software will then recognise that this is a TTTR file and offer to transform it into histogram data. This can then be copied to the clipboard or saved as a *.PHD file which can then be used with FluoFit as usual. In this case of course one could have collected ordinary histogram files (*.PHD) in the first place.
Note that we also provide a powerful TTTR analysis package, our "SymphoTime" software. This also provides decay fitting functionality similar to FluoFit.

> I want to use TTTR mode to investigate single molecule blinking. I need the
> the global photon arrival time so that I can further construct a fluorescence-time
> trajectory, which tells the photon counts as a function of time. This trajectory
> should facilitate the observation of, for example, on-off state of a blinking
> molecule or the emission behavior between donor and acceptor in FRET.
> I used T3 mode as the data acquisition mode and saved the data in a
> PT3 file. Then, I run the Matlab script read_pt3.m to open the saved t3r file.
> However, here was nothing plotted and neither can I find the global photon
> arrival time. Please tell me how to get the photon arrival time so that I get the
> photon counts as a function of time.

Indeed there is nothing plotted by the file demo. This is intentional. The TTTR format carries all the information about the photon arrival dynamics and there are virually infinite possibilities of evaluating such data. Because this is application dependent we cannot do this for you. Remember that this script is only a demo for reading the file. The analysis you must do yourself. In order to get at the global times you should take a look at the output file xxx.out that is generated by read_pt3.m. There you see a list of all the arrival times. Then take a look at read_pt3.m and see how it generates this output. The variable you need is called "Truetime".

> Does the PicoHarp have a buffer memory to avoid keeping the processor busy all
> the time? If there is a buffer, what transfer mode do you use?

Yes there is such a buffer. In case of forming histograms (oscilloscope mode, integration mode, TRES) that buffer holds the histogram, which is being created in hardware on the board, completely without the PC's processor.
In TTTR mode we form no histograms but transfer each photon event to the PC as a 32 bit record. In this case the on-board memory is configured as a large FiFo buffer (256k photon records). This gives the PC and the operating system enough time to attend to other jobs without losing data. In this mode the PicoHarp 300 can achieve a sustained count rate of over 5M counts/s, storing the data directly to disk.
The transfer is done in USB 2.0 bulk mode. This permits a theoretical bandwidth of 480 Mbits/s but effective bandwidth depends on the PC and its USB controller (typically max. 30 MBytes/s)

> My PicoHarp 300 is working fine. However, I have been looking at the data now
> and all of my pt3 files have "overflow" counts. I had understood this to mean
> that the photons came too quickly for the electronics.
> I am wondering if it is related to another anomaly in the time-tagged data,
> namely, as I scan the data to identify the smallest time between two
> photons I encounter spots where the photon "macro-time" suddenly goes back
> to smaller values. These data should be monotonically increasing through a
> data file. Could this be related to te overflows?

These 'overflows' are not a problem at all. They are not errors. They are just markers to tell the software that the hardware counter has wrapped around. The software counter can then be adjusted accordingly. The jumps in the time tag are only present in the raw data. The correct and monotonically increasing list of time tags must be obtained by the software that loads and processes the TTTR file. The demo software shows how this is done. One would preferably use a 64 bit variable for this. The reason why it is done this way is this: One would need many more bits in each data record to obtain a reasonably long measurement time without overflow. This would lead to more data to be handled. By simply allowing those overflows,
but marking them as such for later correction, the recording time can be made infinite, yet with reasonable data traffic on the USB.

> If I send identical signals to two PicoHarps, I can I expect the recorded
> signal patterns in TTTR mode to be identical? It seems that I should be able
> to achieve at least accuracy within the specified crystal accuracy.

You need to distinguish T2 and T3 Mode here. Let's start with T2. The crystal then indeed determines the accuracy you can expect, apart from an offset due to different start times. However, since both clocks are indpendent they will drift apart. Crystal oscillators are accurate to typically some tens of ppm, i.e. the error shows in the 4th or 5th decimal place. In T3 Mode the sync accuracy is determined by the sync source. You can give both PicoHarps the same sync signal. Then the readings will be very close because the “fine count” in T3 Mode is not affected by drift. The short range stabiliy of the PicoHarp is very good. Time differences up to milliseconds can still be measured with the full accuracy.

> The PicoHarp manual says: "From the T3 Mode event records it is possible to
> precisely determine which sync period a photon event belongs to. Since the sync
> period is also known precisely, this furthermore allows to reconstruct the arrival
> time of the photon with respect to the overall experiment time."
> How can I do this, given the T3 file I collected?

It is quite easy. First your software needs to correct for overflows in the sync counter. This is best done by keeping an “overflow counter” in a 64 bit integer variable and incrementing it by 65536 each time an overflow marker is found. The overflow corrected sync number trueNsync is then obtained by adding the uncorrected sync number to the current overflow counter. Then you calculate the absolute time as follows:

truetime = trueNsync*Tsync + dtime*Resolution

where TSync is the sync period and dtime is the picosecond delaytime of the photon from the sync, as found in the PT3 record. The whole procedure is shown in the file demos (see e.g. pt3demo.c).

Keep in mind that the absolute time is only as accurate as your sync source is. crystal oscillators always have some ppm of calibration error as well as thermal drift. As long as the sync source is your only timing reference it probably won't matter.

> We would like to use TTTR mode to do FCS. Since we do not need the
> picosecond TCSPC timing, a cw laser would be sufficient. However, then there is
> no sync signal. Is there a way to solve this?

You can surely use the PicoHarp for FCS with a cw laser. For this application you would use T2 mode. In this case a sync signal is not required. In fact, you can even use boh input channels for photon detectors. This allows you to do cross correlations. The highest time resolution is 4 ps, which is much more than you can get with any hardware correlator. We have published a very efficient software correlation algorithm but we also offer ready-to-use software products for this kind of analysis.

> I'm trying to get some insight about peculiar behavior of our PE
> SPADs. Specifically, we have making TCSPC measurements with the
> PicoHarp 300 but have not succeeded with FCS. The characteristic
> behavior is a rapid decay of the correlation function that is over in an
> unreasonably short time. The behavior is sample and viscosity
> independent. I talked about this to someone who had similar problems

> with PE SPADs that he purchased around the same time we purchased ours.
> He has an older PE SPAD that works perfectly. I'm confused.
> Any theories or thoughts or related experience or suggestions?

The problem you are observing is due to SPAD afterpulsing. After a true (photon induced) avalanche, the detectors some times generate secondary avalanches due to residual trapped carriers in the active region. This occurs with a delay of up to (typically) a few microseconds and therefore shows nastily in the correlation curve. Some people were lucky to find (selected, or older) PE SPADS that did not show this behaviour (or very little of it). It essentially depends on the number of lattice defects in the active junction area. However, as this is more or less inherent to any SPAD, the profound way of avoiding it is to use two detectors and doing a cross-correlation between them. In case of the PicoHarp you can do this very easily in T2 mode.

Routing and External Device Synchronization

> What is possible in terms of routing the data signals from multiple detectors into
> different parts of the histogram memory?

You can use a router (dedicated for the PicoHarp). We offer them for different detector types and numbers. In histogramming mode the counts from the different detectors will then be collected in separate memory blocks. In the TTTR modes the photon records will have their input channel code bits set according to the routing information.

> Could I fill different histograms off of the cards internal clock or an external
> trigger? I would like to collect decays at 1 µs to 1 ms time intervals without gaps.
> Similarly, could I fill different histograms off of a scanner signal (so I can store
> a histogram for each pixel)?

This is essentially a continuous histogramming mode with an internal or external clock. We provided this with earlier products but meanwhile it has been fully replaced by TTTR mode with markers. The latter is much more flexible and more efficient. Consider how many photons you can possibly expect in a microsecond. Fast continuous histogramming creates mostly empty histogram bins. This inflates the data transfer load beyond the necessary. If you just need histograms over fixed small time slices it is easy to obtain them from the TTTR data stream. It even can be done in real-time. You also have ultimate flexibility in varying the time slices or even adapting them to the data (e.g. photon bursts). If you need to set your histogram boundaries by external signals you can use the external marker signals (TTL). Each time a marker edge is detected, a corresponding record is inserted in the TTTR data stream. There are three such TTL inputs, each with programmable edge polarity. This is perfect for imaging applications. You just assign pixel, line, and frame clocks to three of these markers. You may not even want to use the pixel clock if it is too fast. Because there is a time tag on each record you can actually calculate the corresponding pixel from the time to the line sync. There is another signal that can be used to blank out data collection e.g. during flyback. At the end of the manual there is a description of the control port connectors and all these signals.

> Your "router" devices for several detectors are of interest to me. I would like to
> use 4 detectors. Do these devices allow to record the 4 channels at once,
> or do they only allow switching between the 4 channels, i.e. you look at the
> 4 different channels one at a time?
> In opposite terms: Can the stops from the separate detectors be interspersed
> and can one tell which detector they came from? Also, does this option change
> the time resolution of the PicoHarp?

The routing principle is 'signal driven multiplexing', i.e. the 4 router channels share one timing circuit on a 'first come first serve' principle and they can be interspersed. This works fine because the probability of receiving a photon in one sync period is very low. It must be kept on the order of 1% anyway, which is a prerequisite for avoiding pile-up in TCSPC. If two photons arrive during the same sync period, only the first one will be accepted.
In the routers there is no round-robin-like switching between the channels. The channels are all 'open' at the same time. The router recognises in which channel it received a photon and delivers that information to the PicoHarp via a dedicated control port. The next photon can be on any other channel, whichever comes first.
Using the router does not change the time resolution. Of course, any additional element in the signal path can increase noise and RF pickup but by reasonable standards it is not significant.
The only limitation the routers imply is that the total count rate the PicoHarp can handle must be shared by the router channels. In order to keep crosstalk low we recommend not to exceed a maximum of 500.000 counts/sec per router channel.

> Regarding the routers: Does the multiplexing cause losses?
> I.e. does it have consequences on the number of events per channel?

The routing scheme is a signal driven multiplexing. The 4 channels must share the throughput of the card, i.e. if The PicoHarp is in its dead time due to a photon it received one one rouer channel, it will be blind for a photon on another router channel. This is usually not a problem due to low count probabilities enforced by single photon statistics anyway. In fact the counting efficiency of each channel is only minimally reduced, while the overall efficiency with four channels is actually increased!

> I understand that with the router, the data of multiple photon channels can be
> taken (with common SYNC) and T3 mode can be used to collect data for FCS.
> However, I read that FCS is then only good for lagtimes greater than the deadtime.
> Why is that?

In case of a router, the crosscorrelated channels must share the same time digitizer channel of the PicoHarp. If it is busy with processing one photon it cannot accept another before it is finished (deadtime). Therefore, you never see photon coincidences at a time shorter than the deadtime. This means your FCS curve will begin only just above the deadtime (~90ns).

> What happens if two router channels receive a signal at the same time?
> Is there a deadtime?

If two signals arrive at the same time they are both discarded because a clean routing decision is then not possible. The probabiliy for his is very low.
Apart from the PicoHarp's deadtime, the router also has some kind of deatime. The router uses a 60 ns time window to make the decision as to whether the count is valid. This is the case if during that period only ONE channel received a signal. This 60 ns period is to be considered a deadtime, however, it is mostly masked by the PicoHarps own deadtime of ~90 ns.

> For test purposes I connected my signals directly to the PicoHarp, i.e. bypassing
> the router. Now I get no counts when I start a measurement, even though the
> sync rate and CFD rate look allright.

You probably left the router connected to the PicoHarp via the Sub-D cable.
In that case the router always pulls the "valid" signal (count enable) low and so the PicoHarp cannot collect counts. You need to disconnect the router if you don't want to use it.

> In addition to an input of the PHR 402 router I would like to feed another
> 50 Ohms load from the same source (Perkin Elmer SPCM). I use Tee splitters
> that are proper 50 Ohm splitters. I noticed that I get strange signals and many
> more dark counts when I connect the signal from the Tee.

From 50 Ohms power splitters you will get out of each leg only half the original amplitude. This is not sufficient for triggering the TTL router properly. If you turn the discrimination level very low, this might be a reason why you see more spurious counts. So you can connect the SPCM only directly to the router.

> Our main application of interest is synchronizing the PicoHarp
> acquisition with a scanner. The imaging works as follows: We move the
> scanner, stop, acquire data, and then move the scanner again. We would
> like to start the PicoHarp acquisition after the scanner stops, and stop
> the acquisition before the scanner starts moving.

The easiest way of doing this is just by software. There is a driver library (DLL) that you can use to initiate each measurement and fetch the data when done. This can be repeated in a loop over all your image pixels. Of course this is relatively slow due to software overhead and unpredictable delays by Windows itself. Faster and more precisely synchronized scanning requires hardware signals. (See the manual and the topics here about Markers in TTTR).

> What synchronization signals can the PicoHarp utilize? I see 2 SMA
> connectors (for start and SYNC I guess) and then two D connectors as
> well. What is available on them? Can the arriving photons be routed by other

> external parameters?

The golden SMA connectors are the two timing inputs of the PicoHarp. In the most common histogramming mode, as well as in T3 mode, they are used for a sync and a photon detector signal. In T2 mode they may both be used for detectors.
The D connectors are for routers and external experiment control. For instance, some pins accept TTL signals for external markers etc. Other pins provide +3.3 and +5V for external devices (limited current). Most notably the "RT" connector is used to connect the PHR 40x routers or other future extension modules. At the end of the manual there is a description of the control port connectors and all their signals.

> When I time-stamp photons in TTTR mode, I would like to synchronize other
> parts of the experiment to the timing of the photons to the microsecond level.
> I could conceivably use the external event tagging to do this, but I in general
> prefer to have everything running off the same clock. Then the synchronization
> is down to the level of the clock frequency. Is there a clock output available?
> Similarly, is there an output trigger when the acqisition starts?

There is no clock ouput. In T3 mode it is probably best to synchronize everything to the sync input. However, there is a signal on the control port when the TTTR run starts (time tag Zero). At the end of the manual there is a description of the control port connectors and all their signals.

> How is a scanner (e.g.piezo stage scanner for a microscope) interface
> done? Can the PicoHarp clock the scanner, or does it need an input from
> the scanner?

The scanner needs to provide synchronization signals (TTL). The PicoHarp allows to insert external markers in the TTTR data stream, so that you can synchronize to virtually anything, including laser scanning microscopes (LSM). In that case the markers will represent line and frame sync pulses.

Antibunching and Coincidence Correlation

> Can the PicoHarp measure antibunching? How is it done?

Yes, the PicoHarp can measure it. How it is done depends on the process you want to observe. The physical background of typical (fast) antibunching experiments is that a emitter (e.g. a single molecule) can only be in an excited state or in the ground state. If it is already excited it cannot be excited again until it has returned to the ground state. What follows is that the molecule can only emit one photon at a time when it is in the excited state. This can be used e.g to measure the average time in the excited state, this is identical to the fluorescence lifetime.
One typically uses two detectors that each detect some of the photons from the cw excited molecule. The histogram of time differences will have a dip reflecting the excited state lifetime. The creation of such a dip is basically what reflects antibunching. The histogram would be a flat line if there were just random illumination of the detectors i.e. no single molecule.

In the experiment one typically uses two photon detectors, one at the start input (Channel 0), one at the stop input (Channel 1) . The data can be collected in histogramming mode or T2 mode. Histogramming only captures subsequent evens on the two channels while T2 Mode allows to calculate full correlation functions. The excitation light is usually cw. It can also be pulsed, which will result in a comb structure.

> Can you provide a reference from users who used the PicoHarp for
> antibunching experiments?

The PicoHarp is a relatively new product, so publications are only just beginning to appear. However, the list is growing rapidly, so please check our bibliography frequently. It is at

http://www.picoquant.com/_biblio.htm

Of course we conduct important experiments also in our own lab, often with users who come to try their samples with our instruments. Anitbunching experiments are routinely performed with our MicroTime system.

> How can I get negative(-) time delay information to get the symmetrical anti-
> bunching signal such as shown in the literature an on your web pages?

You need to insert enough cable delay in channel 1. That will shift the histogram into the center.

> We're trying to do a Hanbury Brown-Twiss coincidence measurement with
> the PicoHarp. We've done this many times before with the TimeHarp, but
> today was our first try with the PicoHarp. We found that we have to
> insert a long (~25 meter) coaxial delay into the Channel 1 (start)
> input, in order to put the true zero delay at a modest positive delay
> (~100 ns). I would have expected we would need to delay the sync
> (Channel 0) input, not the start input. Am I missing something
> obvious? We checked on a scope with a very low rep rate (kHz) laser,
> and there are no other large delay differences in the optical or
> electrical paths from each detector in the coincidence measurement.

Your observation makes perfect sense. The Picoharp indeed has its histogram time zero at almost exactly zero delay of the electrical inputs. You need to delay the Channel 1 input because the PicoHarp works in forward start-stop mode, as opposed to the TimeHarp that required reverse start-stop. In this sense your terminology is wrong. Channel 1 is not the start input.
If you want to avoid the cable delay you could use T2 mode. It gives you individual event times and you can calculate any time differences within or across the channels. That would allow you to work with negative times. Of course the data is not so nicely visually accessible. You would have to write a program for it. The advantage is that you can calculate not only coincidence correlaions between pairs of consecutive events but also true cross- and autocorrelations.

> I know that with the TimeHarp, the true zero delay between the input signals
> appears at a large positive delay on the TimeHarp software's plot. This
> makes sense to me, since it's a reverse start-stop configuration - so
> signals occuring near the true zero delay will end up at the very end of
> the time window, which is flipped before plotting. With the PicoHarp,
> the true zero delay seems to be shifted so it appears much closer to t=0
> on the PicoHarp plot. (Of course, this is very useful, and makes data
> taking much easier.) What does the PicoHarp do differently than the
> TimeHarp to shift the data this way? Perhaps if I understand this I can
> better understand why I need to delay the start in the coincidence
> measurement.

Indeed the PicoHarp is quite different from the TimeHarp. The TimeHarp is only a "stopwatch" i.e. it always needs pairs of start and stop. The limited speed of the time digitizer requires reverse start-stop operation as well as mutual gating of the start and stop inputs, which in turn causes the non-symmetric delay characteristics of the two inputs.
With the PicoHarp we tried to put in all the things we learnt over the years from customer's requirements. At the same time we also improved the circuit technology. Instead of a stopwatch we basically implemented independent but synchronized picosecond "watches" for each input channel. The relative timing for the histogramming mode is done by simply subtracting the two watch readings. I.e. the internal raw data is like that of T2 mode. The histogramming is implemented in FPGA on op of that.
Knowing that there are just two independent and identical watches whose times get subtracted the "true zero at t=0" will be obvious to you. Small residual offsets (typically <1ns) may of course still remain due to small differences in the two CFDs or slightly different pulse rise times, level settings etc.
Having the two independent channels also completely changed the story of "reverse start stop mode" which is no longer necessary! Indeed it always was just a way of overcoming instrument limitations and often led to confusion due to the reversion in time. The only thing we needed to solve was that of fast syncs (>10 MHz) from laser sources used in classical TCSPC, so that they would not be lost in dead time. This was done by implementing the sync divider in channel 0. However, the divider should not be used for coincidence correlation experiments between photons from two detectors.

> For an anti-bunching experiment, do you have any suggestions to measure the
> delay offset caused from the two different setups of detectors (such as different
> lengths of optical paths)?
> I am using two APDs to measure the time delay between the two successive
> photons out of my system.
> The emission is divided up and introduced to the two different APDs.

The absolute time delay is to some extent predictable but you can only measure relative delays. It depends on your detectors, optical paths and cables. The PicoHarp also has some (constant) internal delays but they are very small, typically well under 1 ns between the channels. Therefore, if your optical and electrical paths are identical, the temporal coincidence of events on the two detectors will apper near the histogram time 0. If you are unlucky it may be slightly in the negative, so you won't see it. You can then either swap the two inputs or insert some cable delay.

One way to solve this empirically is to try and vary your cable lengths until your delay is right to shift the zero point into the measurement window. There are delay switch boxes that help to make this job easy. It helps to use a wider range at the beginning and then narrow it down. You could illuminate both detecors with the same pulsed light source of long period. This should help to identify the histogram peak at time zero.

> We are using the PicoHarp to measure photon correlations (Hanbury Brown
> and Twiss type setup with the source beam split 50/50 and each branch
> focused on one APD). To determine the position of the zero, we use the signal
> from a single APD (white light), split it witha T-pad, and use the branched signals
> as start and stop. We see that when we add enough cable delay on the
> stop channel, we get a very sharp peak that shifts to the right (towards longer
> times) as we increase that delay.

That makes sense. Make sure you are using a reflection free T-Pad, otherwise you get reflections and unpredictable results.

> Because of optical path difference between the two APDs (assuming that each
> responds electronically within similar time) we expect the "zero" of the two
> APDs to deviate by only a nanosecond or so from the "zero" measured with just
> one APD.

Yes, that's how it should be.

> Using light from a pulsed laser, we get relatively broad peaks separated by the
> repetition period of the laser.

OK, that also makes sense. When you say 'relatively broad' that probably reflects that now the timing uncertainty of the APDs comes into play. If there is a fluorescing material involved you probably also see the effect of the fluorescence decay.

> Having tested the set-up as described above, we still cannot observe
> anti-bunching (light from a single quantum dot under pulsed excitation where
> we expect no peak at t=0).

Well, it should work. We have tested it many times with single molecules. Generally there is no need for pulsed excitation. You should also see a dip with cw and that is easier to verify. Make sure you are suppressing the excitation light sufficiently. Also, make sure you don't get other stray light in. You will probably need a fairly long measurement time because the probability of usable photon pairs is low.

> I am trying to do an antibunching experiment with semiconductor quantum dots.
> I use 2 APDs and a 50/50 beam splitter.
> The problem I have is that the histogramming rate is much lower than
> the two count rates, about 1 order of magnitude. The count rates are both
> 10-20 kHz. Can you suggest what is causing this?

That is quite normal. The input rates are obtained by simple counting
of every event at the respective input. The histogramming rate is the rate of useful
start-stop-pairs that were successfully converted and stored in the histogram.

Since you have random input signals the probability of two start and stop events to pair (so that they fall in the usable measuement span) is rather small.

Software and Installation Questions

> We have installed the PicoHarp 300 to our computer successfully
> but every time I execute the PicoHarp software, I see an error message as follows:
>
> “Could not open PicoHarp USB device.
> Is it connected and switched on?
> Has the device driver been installed?”
>
> If I click the “ok” button, the main window is started. But that is all.
> Most buttons are grey and starting a histogram is not working.

This is the normal behaviour if driver or device are not present, as stated in the manual. The PicoHarp needs a driver to work properly. This error message says that the driver is not installed properly or no PicoHarp is connected.
How did you install the driver? Normally Windows asks for confirmation of the driver installation when the new device is found for the first time. If you installed the software as described in the manual the driver should be there. Did you do that?
If not, the driver is probably not installed. Please disconnect the PicoHarp, uninstall the Picoharp software and repeat the software installation according to the manual. Then re-connect the PicoHarp. Windows should detect the device and install the driver, probaly you are asked to confirm. Then look for the correct installation in the Device Manager. First check if 'PicoHarp 300' is present under the USB tree. There should be no red or yellow marks on it.

> We have installed the PicoHarp 300 to our computer and it is showing correctly in
> the Device Manager. However, every time I start the PicoHarp software, I get an
> error message:

"PicoHarp USB device is not running in USB 2.0 High Speed Mode.
Attached to USB 2.0 port?
Correct mainboard drivers installed?
USB 2.0 High Speed cables (max 5m) used?"

>What does it mean?

The PicoHarp requires a USB 2.0 high speed connection. If such a connection is not established you get this error message. There may be several reasons. First check if your computer supports USB 2.0. Only computers sold from about 2003 and later have it most likely built in. Even then, not all the USB sockets may support USB 2.0. You need to make sure the one you use for the PicoHarp is a USB 2.0 port. Furthermore, wih earlier versions of Windows, USB 2.0 support may require special drivers from the mainboard manufacturer. Finally, not all USB cables are suitable for USB 2.0. You need to make sure they are marked for USB 2.0 high speed. Also, the cable length must not exceed 5 metres. For reliability it may even be better to stay at

3 metres max. If you connect the PicoHarp through a USB hub it must be suitable for USB 2.0. It is best to avoid hubs.

> How can I check my PC for High Speed USB (USB 2.0) Support?

Open the Windows Control Panel, Click on System. Click on the Hardware Tab. Open the Windows Device Manager. Click on the Universal Serial Bus Controllers Folder. If this shows an *ENHANCED* USB Host Controller, the system has High Speed USB (USB 2.0) capability.

> Where are the file access demos? I cannot find them on the CD.

The demos are not directly accessible on the CD. They will be installed by the PicoHarp software setup. After running setup.exe you will find them in a subfolder 'filedemo' under your chosen installation folder.

> I would like to write my own programs to control the PicoHarp.
> On the CD I found the PHLib programming library and several program demos.
> However, when I try to run them I always get error messages.

The programming library is included on the CD but in order to use it you need to purchase a license. You can order it any time and we send you a license upgrade by email.

> Can you recommend a compiler for the programming demos you provide?
> Do we need to buy Microsoft Visual C++?

The C/C++ demos we provide (both for use of the DLL and for reading data files) are tested with the Microsoft compiler but by no means limited to it. You can also use Borland C++ 5.x / C++Builder 3.0 (Win 32 bit). Borland even offers a free command-line version of their compiler. There is also a free GNU compiler environment for Windows called MinGW that works fine (we tested v. 2.0.0-3). It is available at http://www.mingw.org.
